BITS Pilani

Data Management For Machine Learning

innovate achieve lead

BITS Pilani

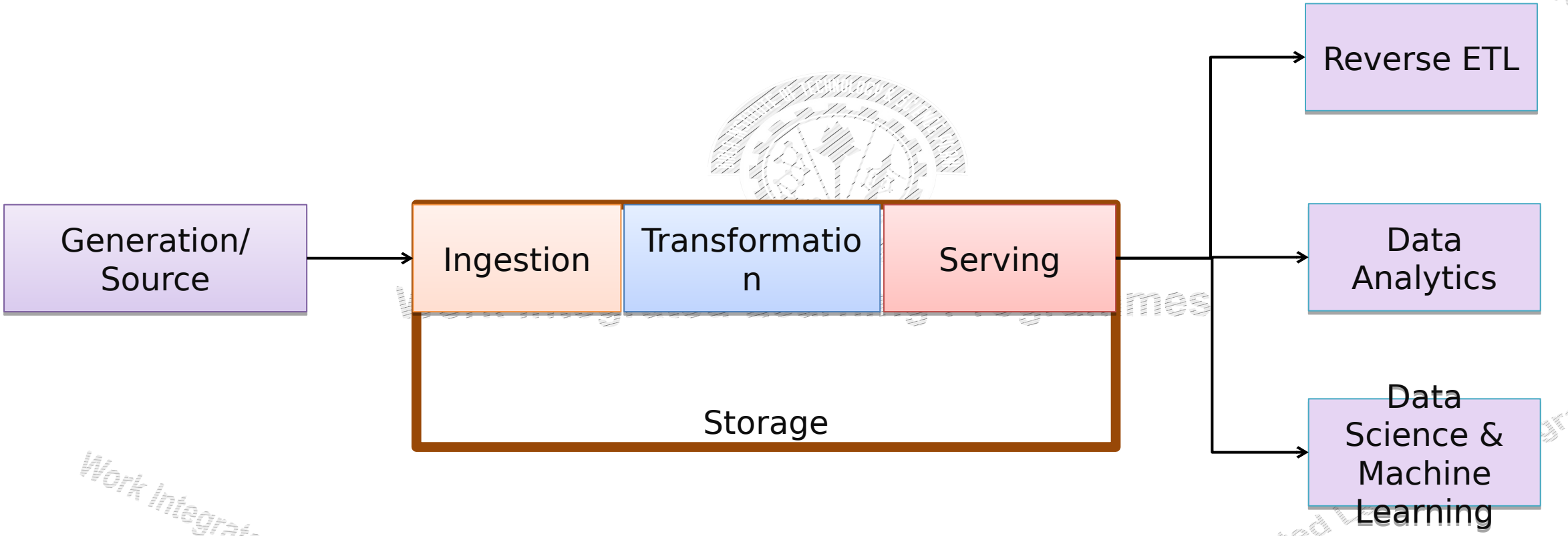
Data Management For Machine Learning Lecture No.5 - Modern Data Infrastructure and DataOps

Agenda

- Introduction
- Data Ingestion Infrastructure
- Data Transformation Infrastructure
- Workflow Orchestration Platforms
- Data Storage Infrastructure
- Data Serving Infrastructure
- Automation in Data Infrastructure Management
- DevOps & DataOps Foundation
- Cloud Foundation
- Security Foundation
- Continuous Training (CT) and Continuous Deployment (CD) pipelines
- DataOps and Observability
- Case Study

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Data Pipeline Architecture – Re-Introduction



BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Key Drivers from Last 2 Decades

- emergence of the major public cloud providers (AWS, GCP, Azure)
- ease of building and deploying
- drop-in storage costs
- emergence of highly scalable, columnar databases (Redshift, Snowflake, BigQuery)

Contributed To

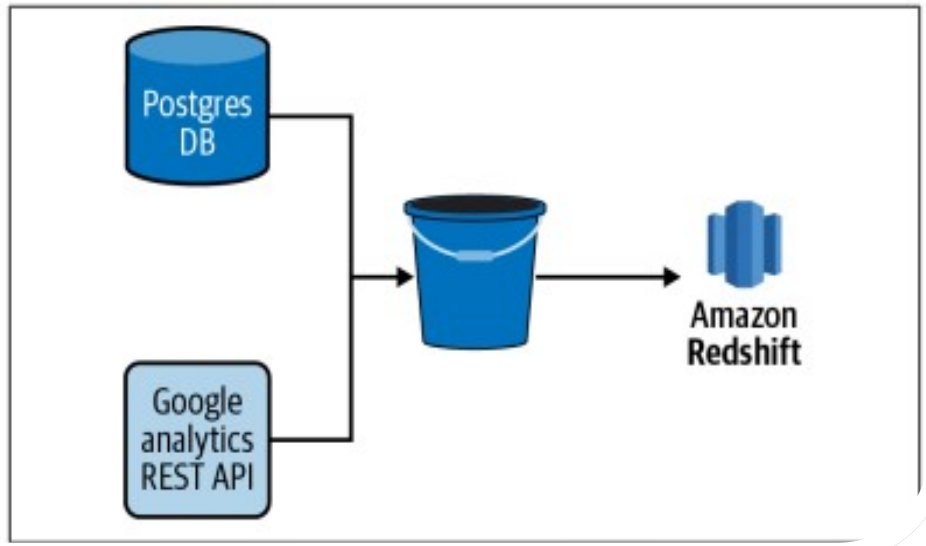
- Modernization of Datawarehouse
- Data Lakes

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Data Ingestion Infrastructure

Work Integrated Learning Programmes

Generation / Source Systems



For instance consider an ecommerce company

- stores data from their shopping cart in a PostgreSQL database behind their web app
- Uses a third-party web analytics tool such as Google Analytics to track usage on their website
- combination of the two data sources is required to get a full understanding of customer behavior leading up to a purchase

- The majority of organizations have dozens, if not hundreds, of data sources that feed their analytics endeavors
- Typical for an analytics team to ingest data from source systems that are built and owned by the organization as well as from third-party tools and vendors
- A data pipeline that ends with an analysis of such behavior starts with the ingestion of data from both sources.
- Regardless of who owns the source data, how you get it and in what form is the first thing a data engineer will examine when building a new data ingestion

Data Ingestion Infra – Key parameters

Interface to the data

- how to get data
- A database behind an application, such as a Postgres or MySQL database
- A layer of abstraction on top of a system such as a REST API
- A stream processing platform such as Apache Kafka
- A shared network file system or cloud storage bucket containing logs, comma-separated value (CSV) files, and other flat files
- A data warehouse or data lake
- Data in HDFS or HBase database

Structure of the data

- JSON from a REST API
- Well-structured data from a MySQL database
- SON within columns of a MySQL database table
- Semi-structured log data
- CSV, fixed-width format (FWF), and other flat file formats
- JSON in flat files
- Stream output from Kafka

Tools

Background

data teams must contend with a diversity of data sources from which to ingest data from

Commercial Tools

A number of commercial and open source tools are available in a modern data infrastructure.

- **Singer**
- **Stitch**
- **Fivetran**

Data ingestion is traditionally both the extract and load steps of an ETL or ELT process

Some tools focus on just these steps, while others provide the user with some transform capabilities as well

Work Integrated Learning Programmes

Data Transformation Infrastructure

Work Integrated Learning Programmes

Transformation & Modelling

Pipelines are also made up of tasks that transform and model data for new purposes

- such as machine learning, analysis, and reporting
- terms often used interchangeably

Data transformation

- **Transforming data is a broad term that is signified by the T in an ETL or ELT process**
- **A transformation can be something**
  - **as simple as converting a timestamp stored in a table from one time zone to another**
  - **a more complex operation that creates a new metric from multiple source columns that are aggregated and filtered through some business logic**

Data modeling

- Data modeling is a more specific type of data transformation
- A data model structures and defines data in a format that is understood and optimized for data analysis
- A data model is usually represented as one or more tables in a data warehouse

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Tools

Data Transformation

- **Some data ingestion tools provide some level of data transformation capabilities, but these are often quite simple**
- **for the sake of protecting personally identifiable information (PII) it may be desirable to turn an email address into a hashed value**
- **For more complex data transformations and data modeling, find it desirable to seek out tools and frameworks specifically designed for the task, such as dbt**
- **Additionally data transformation is often context-specific and can be written in a language familiar to data engineers and data analysts, such as SQL or Python.**

Data modeling Tools

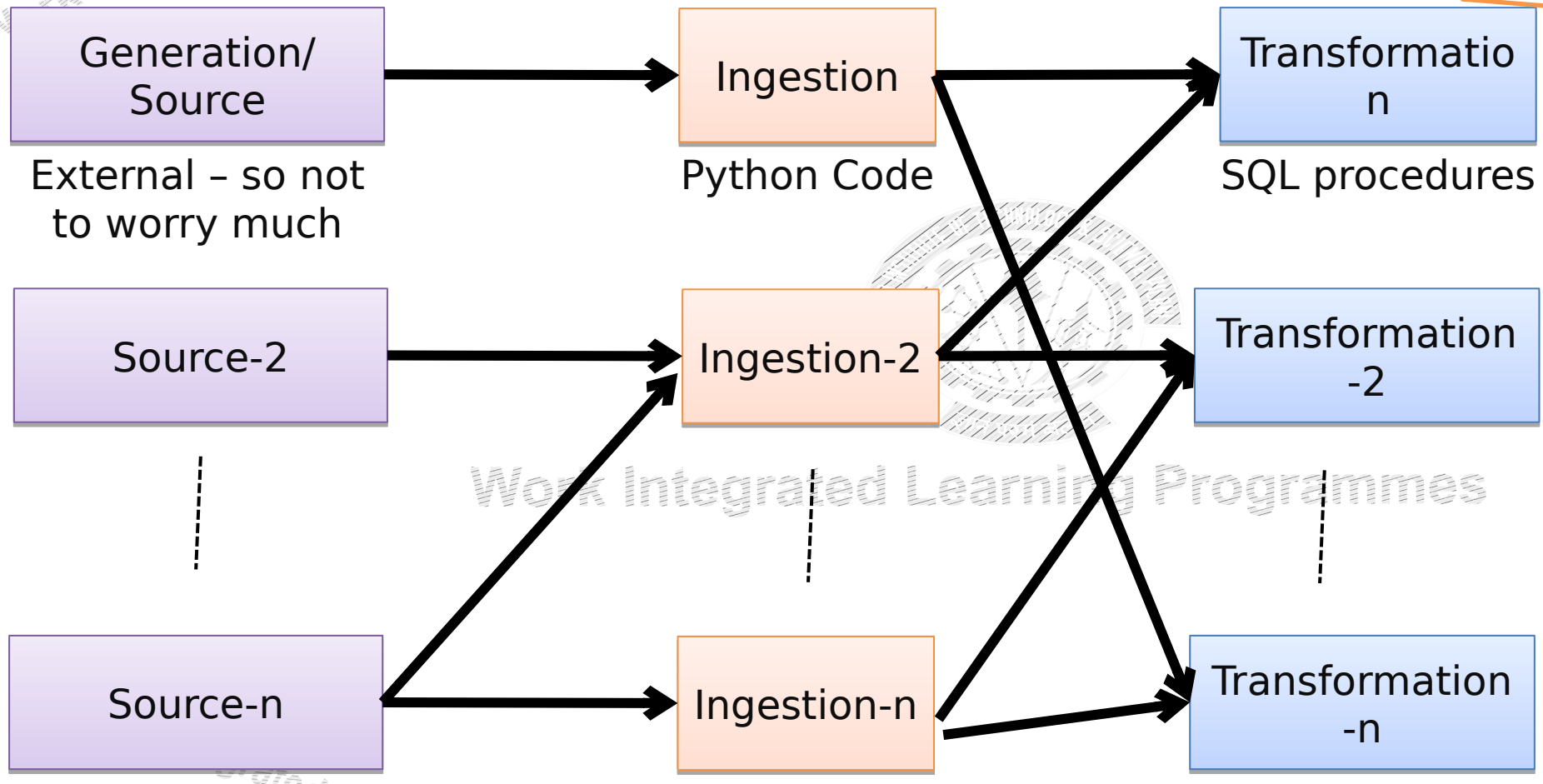
- typically defined and written in SQL or via point-and-click user interfaces
- SQL is a highly accessible language that is common to both data engineers and analysts
  - empowers the analyst to work directly with the data and optimize the design of models for their needs
  - used in nearly every organization, thus providing a familiar entry point for new hires to a team

In most cases, choosing a transformation framework that supports building data models in SQL rather than via a point-and click user interface is desirable!

Workflow Orchestration Platforms

Work Integrated Learning Programmes

Problem Statement



Sounds Simple?

How is this complexity?

What if this has to run multiple times?

How long you can mange this with python & SQL?

University under Section 3 of UGC Act, 1956

Solution

As the complexity and number of data pipelines in an organization grows

- **it's important to introduce a workflow orchestration platform to data infrastructure**
- **manages the scheduling and flow of tasks in a pipeline**

Eg:

- **Apache Airflow, Luigi, and AWS Glue, are designed for more general use cases and are thus used for a wide variety of data pipelines**
- **Kubeflow Pipelines, are designed for more specific use cases and platforms (machine learning workflows built on Docker containers in the case of Kubeflow Pipelines**

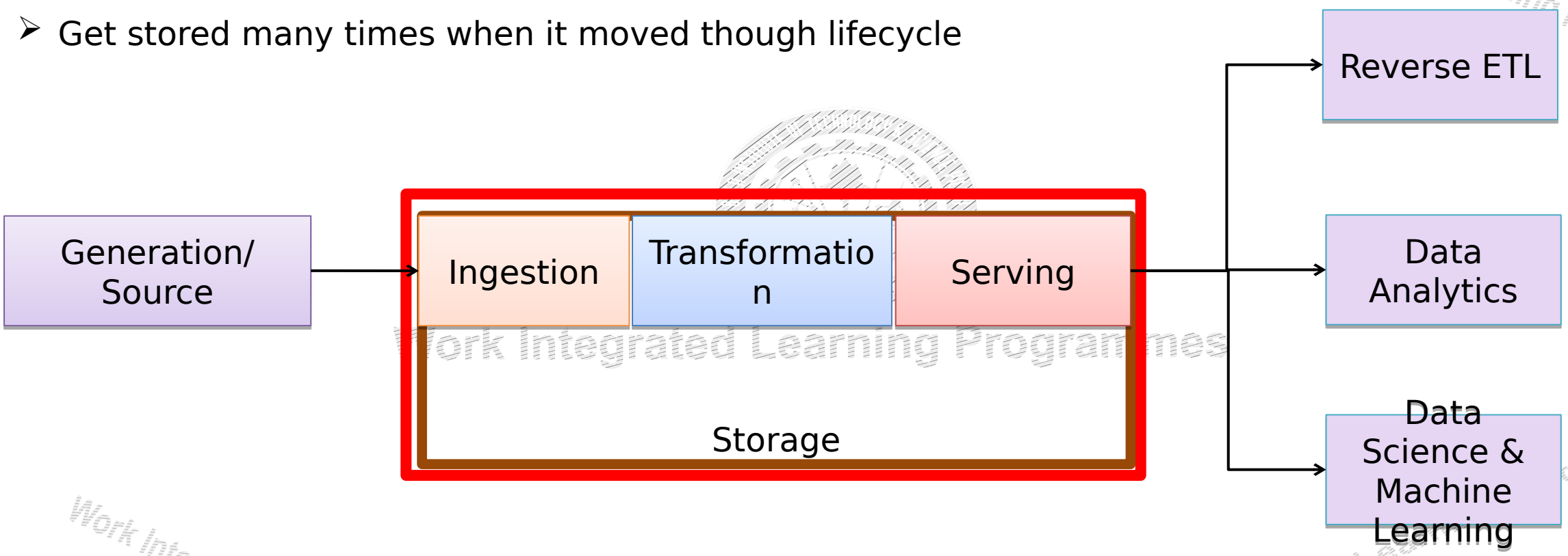
Work Integrated Learning Programmes

Data Storage Infrastructure

Work Integrated Learning Programmes

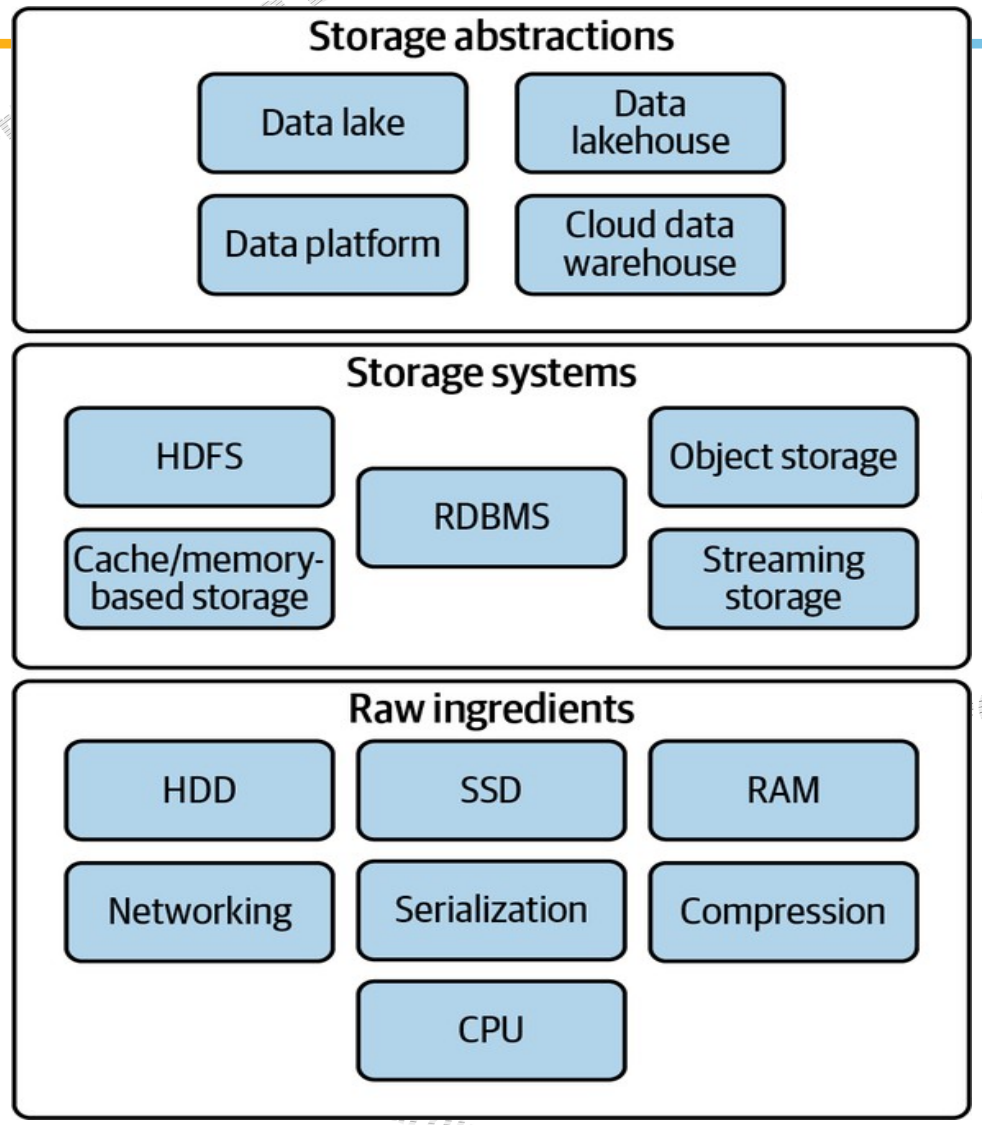
Introduction

- Cornerstone in the data engineering life cycle
- Underlies major stages – ingestion, transformation and serving
- Get stored many times when it moved though lifecycle



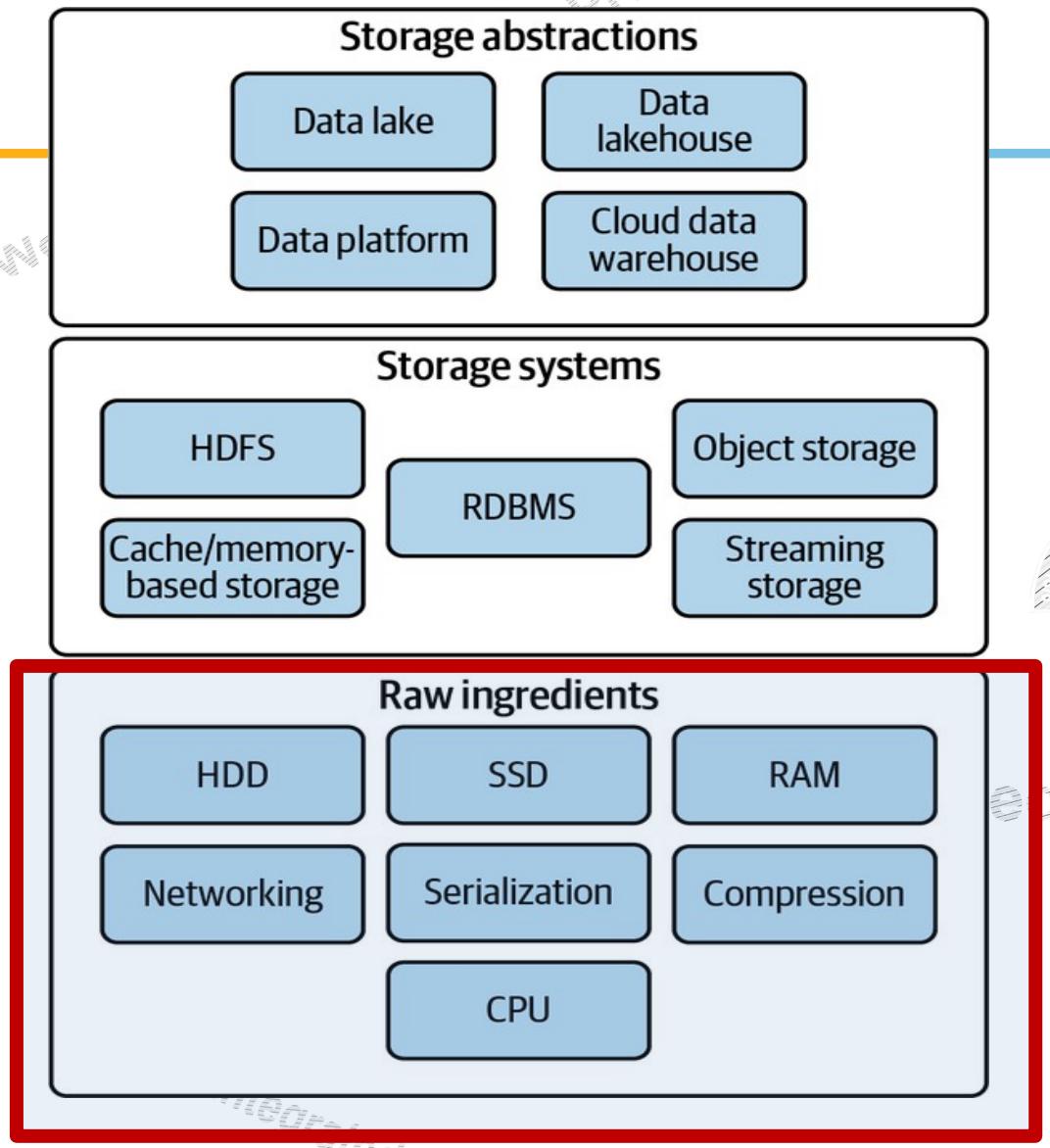
BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Storage Layering



- **Raw Ingredients of Data Storage**: HDD/SDD/RAM, Networking and CPU, Serialization, Compression, Caching
- **Data Storage Systems**: Single Machine Versus Distributed Storage, Eventual Versus Strong Consistency, File Storage, Block Storage, Object Storage, Cache and Memory-Based Storage Systems, The Hadoop Distributed File System, Streaming Storage, Indexes, Partitioning, and Clustering
- **Data Engineering Storage Abstractions**

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956



Data Storage Infrastructure -Raw Ingredients

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

HDD/SDD/RAM

Raw ingredients that compose storage systems, including hard drives, solid state drives, and system memory.

- Hard Drives (HDDs — Hard Disk Drives): These are mechanical storage devices that use rotating disks coated with magnetic material to store data. They offer large storage capacities at relatively lower costs but are slower in terms of data access compared to SSDs.

- Solid State (SSDs): SSDs are storage devices that use flash memory to store data. They have no moving parts, making theDrivesm faster, quieter, and more durable than HDDs. SSDs offer faster data access and are commonly used for faster performance in computers and servers.

- System Memory (RAM — Random Access Memory): RAM is volatile memory used by computers to temporarily store data that the CPU needs to access quickly. It provides fast access to data and instructions, allowing the CPU to operate efficiently. However, unlike storage drives, RAM loses its

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Networking and CPU

- **Networking and CPU are essential components in modern distributed storage systems.**
- **Networking facilitates data distribution across multiple nodes, enhancing performance, durability, and availability.**
- **CPUs handle request servicing, data aggregation, and distribution.**

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Serialization

- Essential for database design and network efficiency, it packs data into a standard format for storage or transmission.
- Purpose: Facilitates data exchange between systems using formats like XML, JSON, or CSV for compatibility across platforms.
- Algorithm: Handles data types, structures, and exceptions, accommodating complex data structures like cyclic references.
- Database Storage: Utilizes row-oriented (e.g., relational databases) and columnar (e.g., columnar databases) approaches, each optimized for specific tasks.
- Most popular current formats (e.g., Apache Parquet), hybrid serialization (e.g., Apache Hudi), and in-memory serialization (e.g., Apache Arrow).

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Compression & Caching

innovate achieve lead

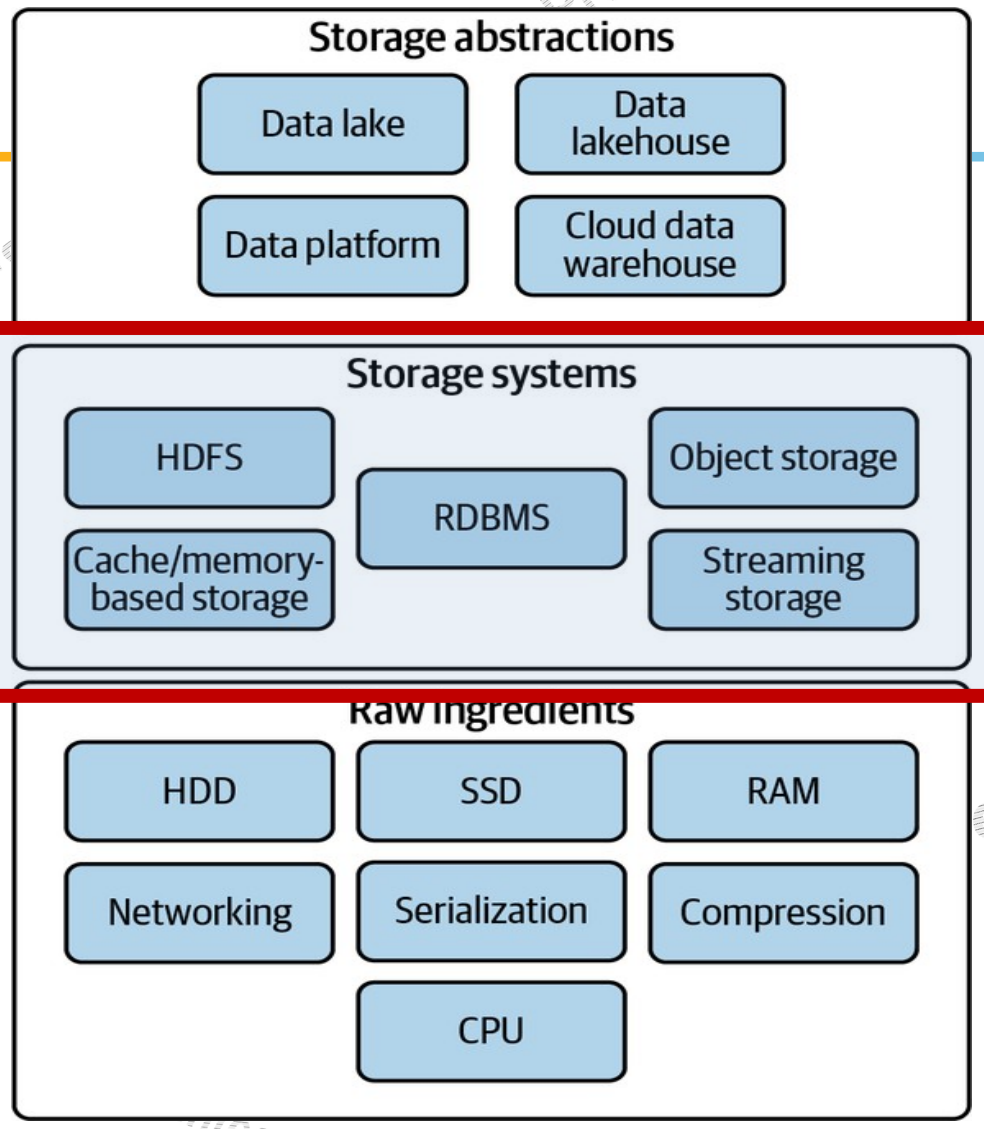
Compression

Critical for storage efficiency, reducing data size and improving scan speed and network performance.

Caching

Stores frequently or recently accessed data in a fast access layer, critical for data serving, processing, and transformation

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

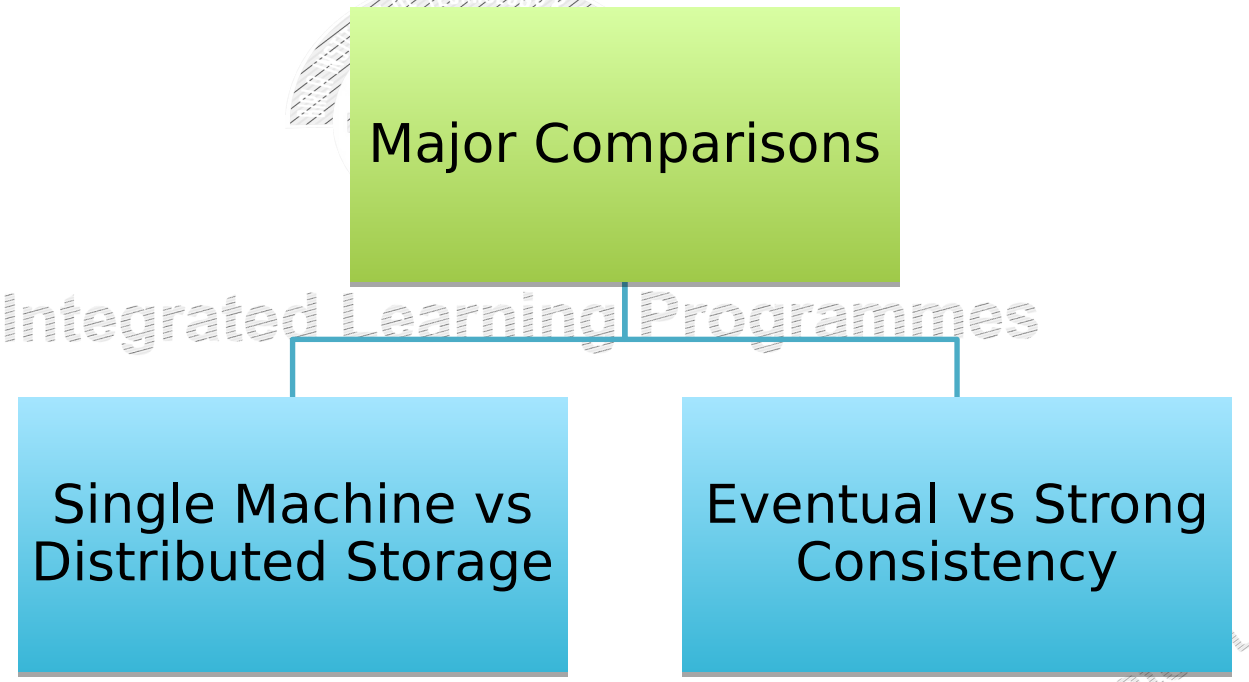


Data Storage Infrastructure -Storage Systems

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

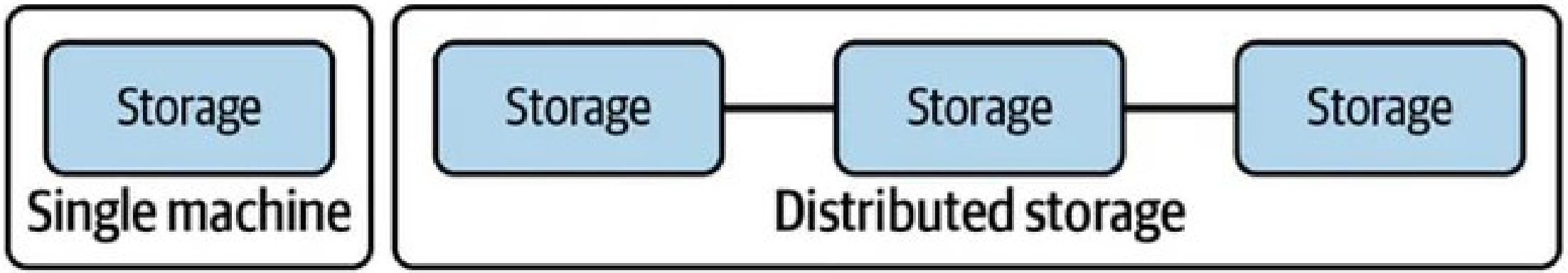
Introduction

- Storage systems exist at a level of abstraction above raw ingredients.
- For exam ple, magnetic disks are a raw storage ingredient, while major cloud object storage Data Storage Systems | 197 platforms and HDFS are storage systems that utilize magnetic disks.
- Before getting into the details, we need to understand major comparisons



BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Single Machine vs Distributed Storage



- As data storage and access patterns become more complex and outgrow the usefulness of a single server distributing data to more than one server becomes necessary
- Data can be stored on multiple servers, known as distributed storage, with purpose is to store data in a distributed fashion
- Distributed storage coordinates the activities of multiple servers
  - to store, retrieve, and process data faster and at a larger scale, all while providing redundancy in case a server becomes unavailable
  - provides built-in redundancy and scalability for large amounts of data
- Examples, object storage, Apache Spark, and cloud data warehouses rely on distributed storage architectures

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Eventual vs Strong Consistency

Eventual Consistency

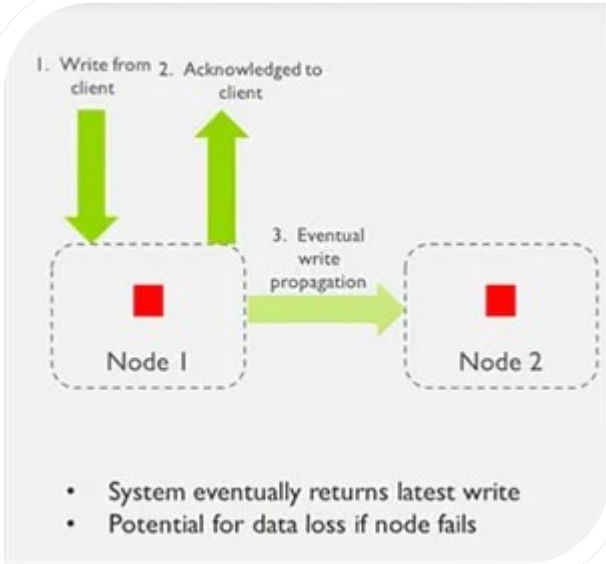
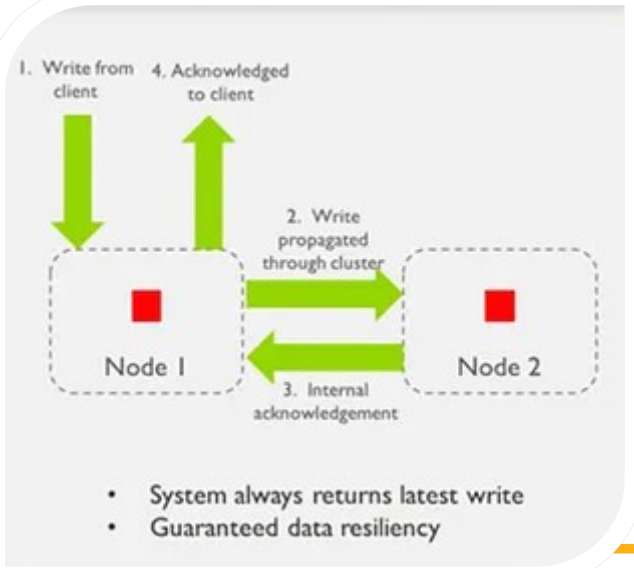

1. Write from client
2. Acknowledged to client
3. Eventual write propagation
Node 1
Node 2
• System eventually returns latest write
• Potential for data loss if node fails

Image Source

Strong Consistency



- A challenge with distributed systems is that your data is spread across multiple servers.
- Unfortunately, distributed systems pose a dilemma for storage and query accuracy.
- It takes time to replicate changes across the nodes of a system; often a balance exists between getting current data and getting "sorta" current data in a distributed database.
- Eventual consistency is a common trade-off in large-scale, distributed systems.
- Eventual consistency allows you to retrieve data quickly without verifying that you have the latest version across all nodes
- With strong consistency, the distributed database ensures that writes to any node are first distributed with a consensus

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Storage Types

Block Storage

Block storage splits data into fixed 'blocks' with unique identifiers. These blocks are stored across a network of storage units (physical and cloud) with no single repository — and are retrieved, very quickly, using specific protocols

Object Storage

Object storage stores various types of data (videos, documents, databases, etc.) inside of a cloud bucket without a hierarchical structure. Data is referred to as an 'object' and comes with the added benefit of metadata and a unique ID.

File Storage

File storage saves data as files stored in folders and subfolders in a hierarchical structure. Files are accessed by following a specific pathway (user\documents\media folder)

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

File Storage

A file is a data entity with specific read, write, and reference characteristics used by software and operating systems

Characteristics of a File

- **has finite length - stream of bytes**
- **supports append operations - append bytes to the files**
- **supports random access - read from any location in the file or write updates to any location**

Organization

- **File storage systems organize files into a directory tree**
- **Eg: /Users/Pravin/file.txt**

File System

- The filesystem stores each directory as metadata about the files and directories that it contains
- consists of the name of each entity, relevant permission details, and a pointer to the actual entity/file
- To find a file on disk, the operating system looks at the metadata at each hierarchy level and follows the pointer to the next subdirectory entity until finally reaching the file itself.

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

File Storage - Types

innovate achieve lead

Local disk storage

- The most familiar type of file storage is an operating system–managed filesystem on a local disk partition of SSD or magnetic dis
- New Technology File System (NTFS) and ext4 are popular filesystems on Windows and Linux respectively
- OS handles the details of storing directory entities, files, and metadata
- designed to write data to allow for easy recovery in the event of power loss during a write, though any unwritten data will still be lost.
- support full read after write consistency; reading immediately after a write will return the written data
- employs various locking strategies to manage concurrent writing attempts to a file

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

File Storage – Types - Contd

innovate achieve lead

Network-attached storage

- provide a file storage system to clients over a network
- prevalent solution for servers; they quite often ship with built-in dedicated NAS interface hardware
- has performance penalties to accessing the filesystem over a network
- significant advantages includes redundancy and reliability, fine-grained control of resources and file sharing across multiple machines

Cloud filesystem services

- provide a fully managed filesystem for use with multiple cloud VMs and applications, potentially including clients outside the cloud environment
- should not be confused with standard storage attached to VMs—generally, block storage with a filesystem managed by the VM operating system
- behave much like NAS solutions, but the details of networking, managing disk clusters, failures, and configuration are fully handled by the cloud vendor
- Amazon Elastic File System (EFS) is an extremely popular example of a cloud filesystem service

Block Storage

- **Fundamentally type of raw storage provided by SSDs and magnetic disks**
- **In the cloud, virtualized block storage is the standard for VMs**
- **allow fine control of storage size, scalability, and data durability beyond that offered by raw disks.**

Block storage applications

- **Transactional database systems generally access disks at a block level to lay out data for optimal performance**
- **Operating system boot disks on cloud VMs**

- **Characteristics of a block**
- A block is the smallest addressable unit of data supported by a disk
- Often 512 bytes of usable data on older disks, but it has now grown to 4,096 bytes for most current disks
- also contain extra bits for error detection/correction and other metadata
- Blocks on magnetic disks are geometrically arranged on a physical platter
- Two blocks on the same track can be read without moving the head, while reading two blocks on separate tracks requires a seek
- Seek time can occur between blocks on an SSD, but this is infinitesimal compared to the seek time for magnetic disk

Block Storage - Types

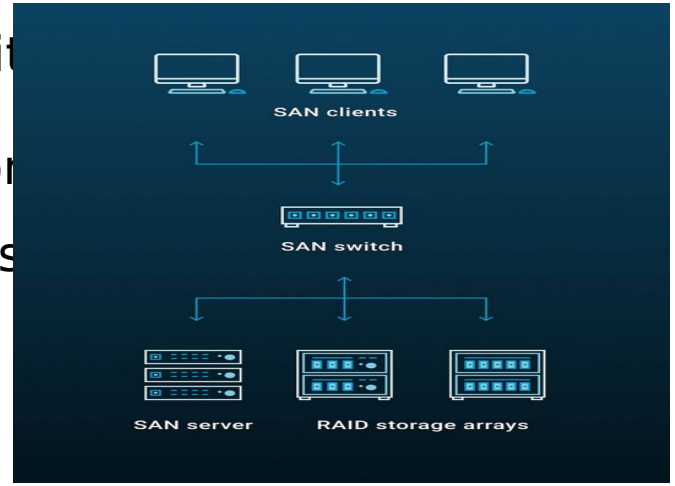
innovate achieve lead

RAID

- **Redundant Array of Independent Disks**
- **simultaneously controls multiple disks to**
  - **improve data durability**
  - **enhance performance**
  - **combine capacity from multiple drives**
- **an array can g system as a s**

SAN

- Storage area network
- provide virtualized block storage devices over a network, typically from a storage pool
- allow fine-grained storage scaling and enhance performance, availability, and durabilit
- commor systems



BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Block Storage – Types - Contd

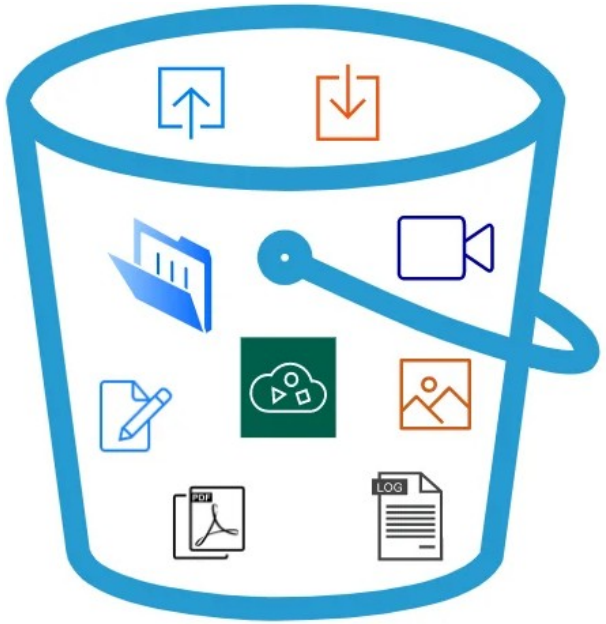
Cloud virtualized block storage

- similar to SAN but free engineers from dealing with SAN clusters and networking details
- Example, Amazon Elastic Block Store (EBS) - default storage for Amazon EC2 virtual machines
- store data separate from the instance host server but in the same zone to support high performance and low latency
- allows EBS volumes to persist when an EC2 instance shuts down, when a host server fails, or even when the instance is deleted
- suitable for applications such as databases, where data durability is a high priority
- replicates all data to at least two separate host machines, protecting data if a disk fails

Local instance volumes

- Cloud providers also offer block storage volumes that are physically attached to the host server running a virtual machine
- generally very low cost (included with the price of the VM in the case of Amazon's EC2 instance store) and provide low latency
- when a VM shuts down or is deleted, the contents of the locally attached disk are lost, whether or not this event was caused by intentional user action
- ensures that a new virtual machine cannot read disk contents belonging to a different customer

Object Storage



- contains objects of all shapes and sizes - a specialized file-like construct —TXT, CSV, JSON, images, videos, or audio
- have grown in importance and popularity with the rise of big data and the cloud
- Amazon S3, Azure Blob Storage, and Google Cloud Storage (GCS) are widely used
- many cloud data warehouses (and a growing number of databases) utilize object storage as their storage layer
- cloud data lakes generally sit on object stores

serverless

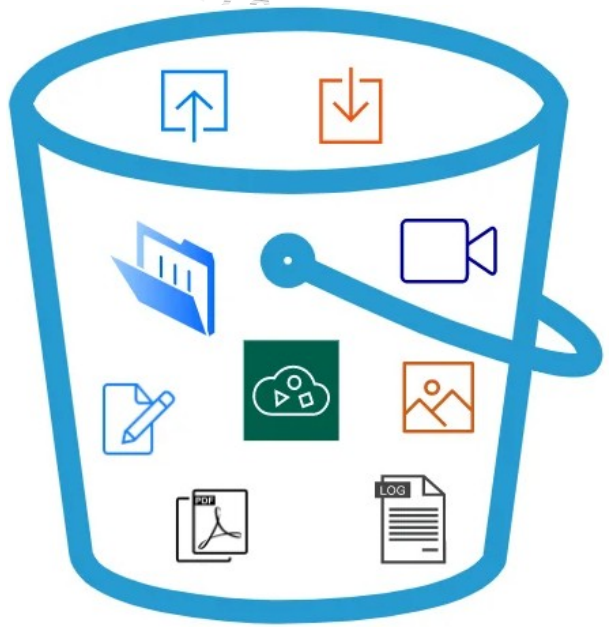
- Object storage was arguably one of the first “serverless” services
- engineers don’t need to consider the characteristics of underlying server clusters or disks

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Object Storage-Contd

An object store is a key-value store for immutable data objects.

- don’t support random writes or append operations; instead, they are written once as a stream of bytes
- After this initial write, objects become immutable
- To change data in an object or append data to it, must rewrite the full object
- support random reads through range requests



Work Integrated Learning Programmes

Cache and Memory-Based Storage Systems

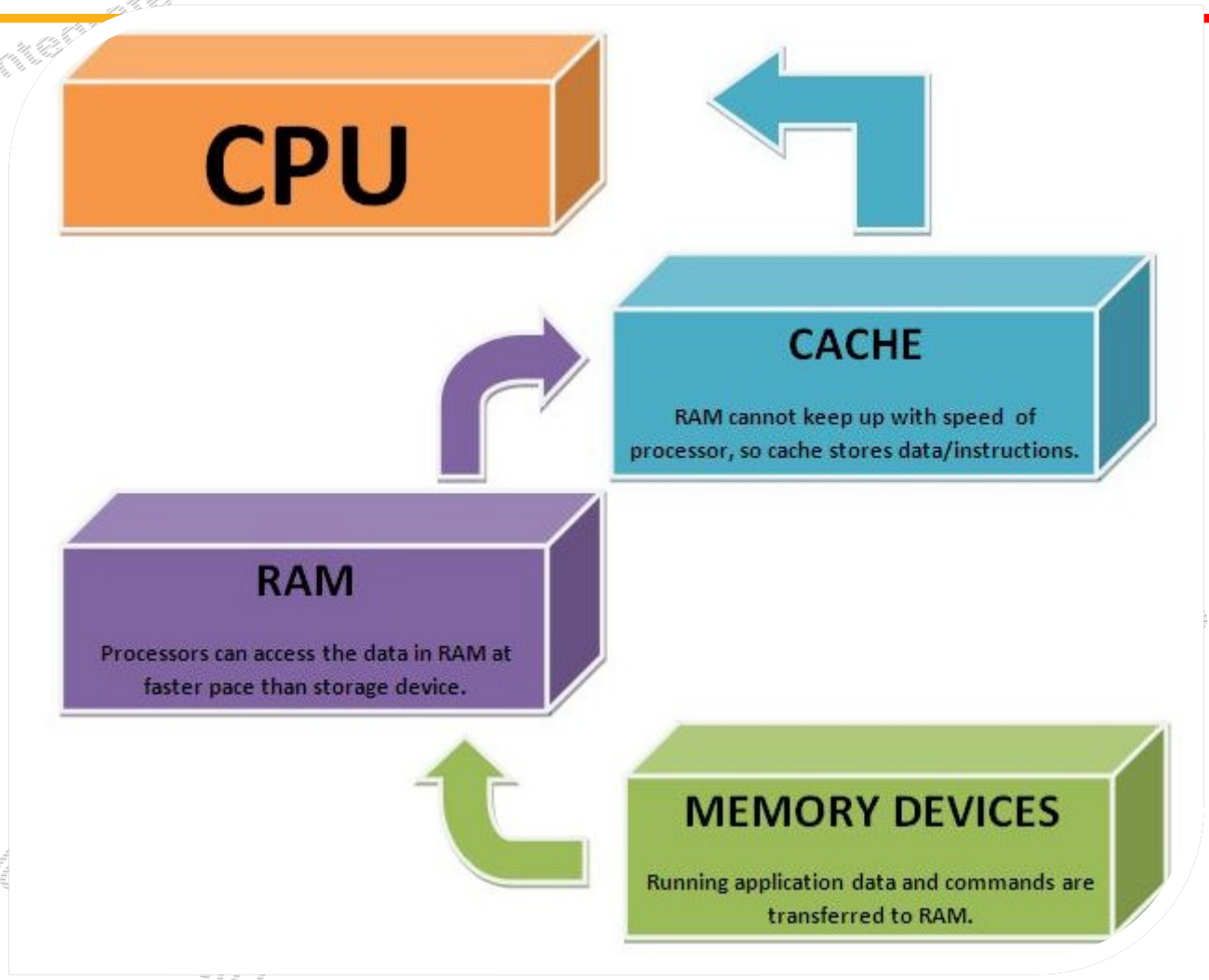

CPU
CACHE
RAM cannot keep up with speed of processor, so cache stores data/instructions.
RAM
Processors can access the data in RAM at faster pace than storage device.
MEMORY DEVICES
Running application data and commands are transferred to RAM.

Image Source

- **RAM offers excellent latency and transfer speeds but extremely vulnerable to data loss**
- **RAM-based storage systems are generally focused on caching applications, presenting data for quick access and high bandwidth**
  - **Data should generally be written to a more durable medium for retention purposes**
  - **These ultra-fast cache systems are useful when data engineers need to serve data with ultra-fast retrieval latency**

Cache and Memory-Based Storage Systems - Contd

Memcached and lightweight object caching

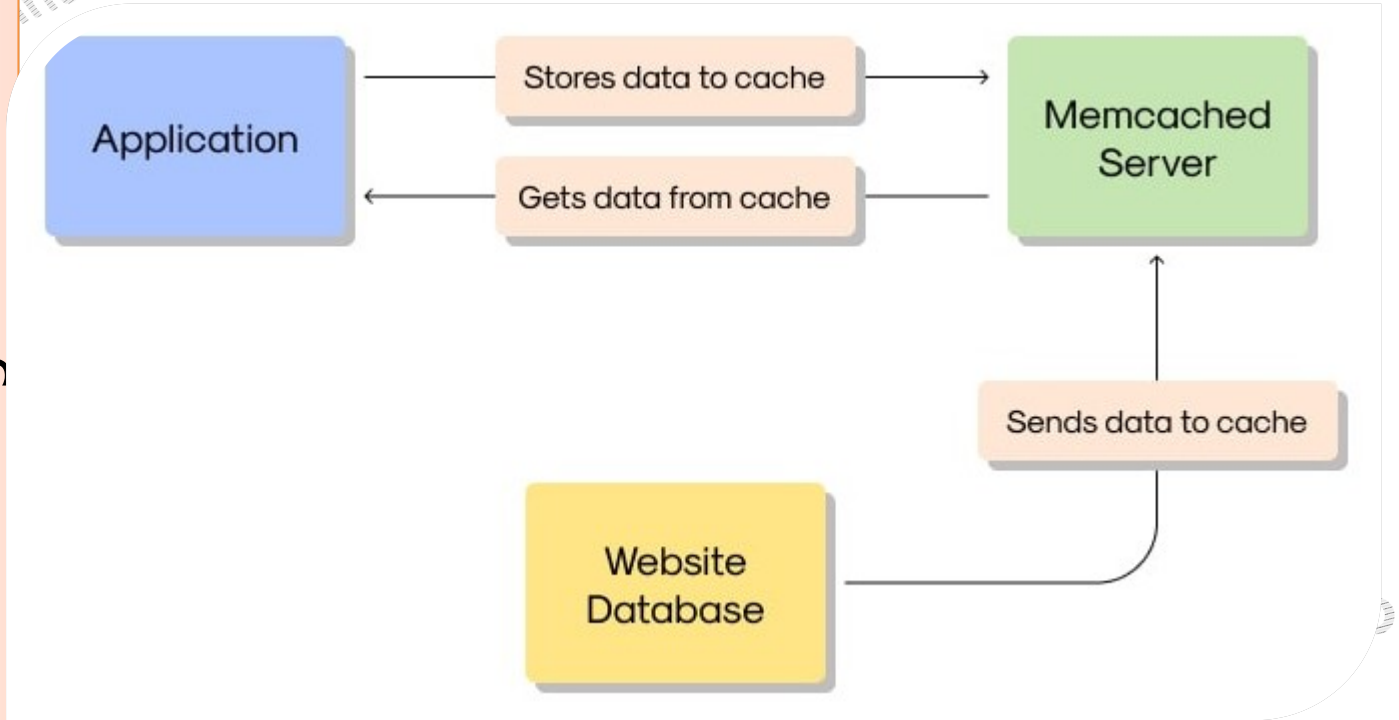


Image Source

- a key-value store designed for caching database query results, API call responses, and more
- uses simple data structures, supporting either string or integer types
- deliver results with very low latency while also taking the load off backend systems

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Cache and Memory-Based Storage Systems - Contd

Redis Cache

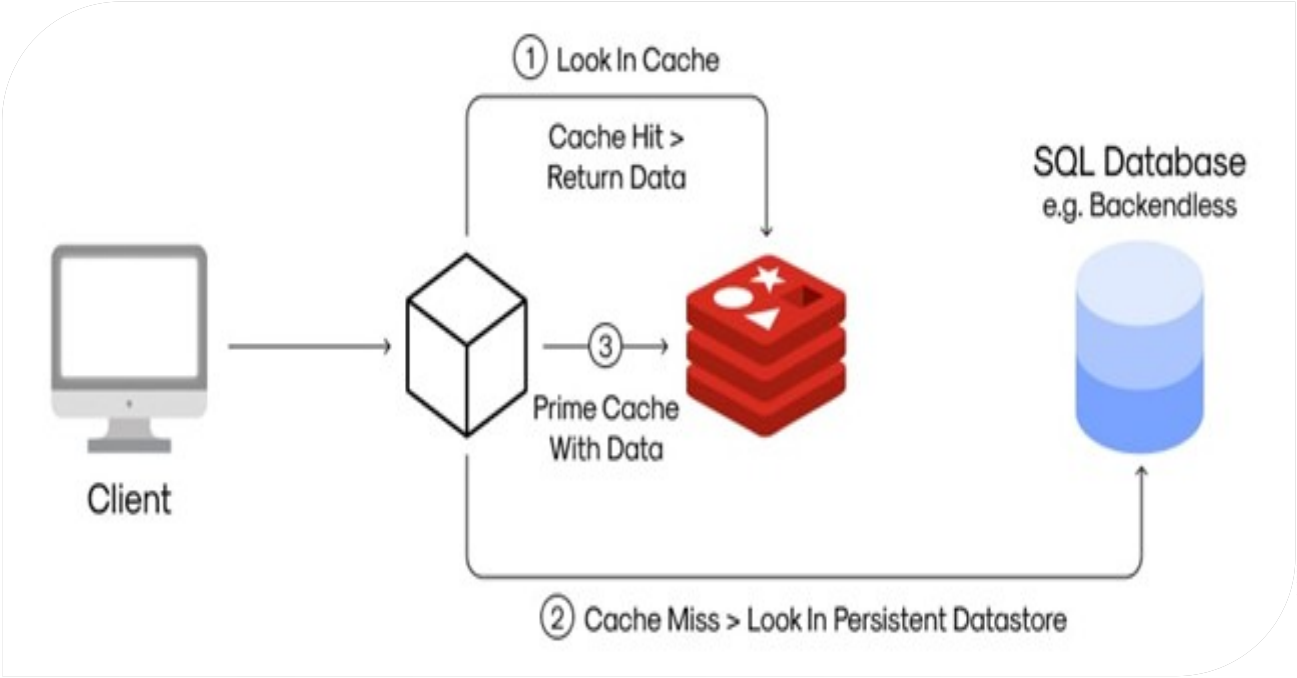


Image Source

- memory caching with optional persistence
- a key-value store, but it supports somewhat more complex data types (such as lists or sets)
- with a typical configuration, Redis writes data roughly every two seconds
- Redis is thus suitable for extremely high-performance applications but can tolerate a small amount of data loss

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Hadoop Distributed File System

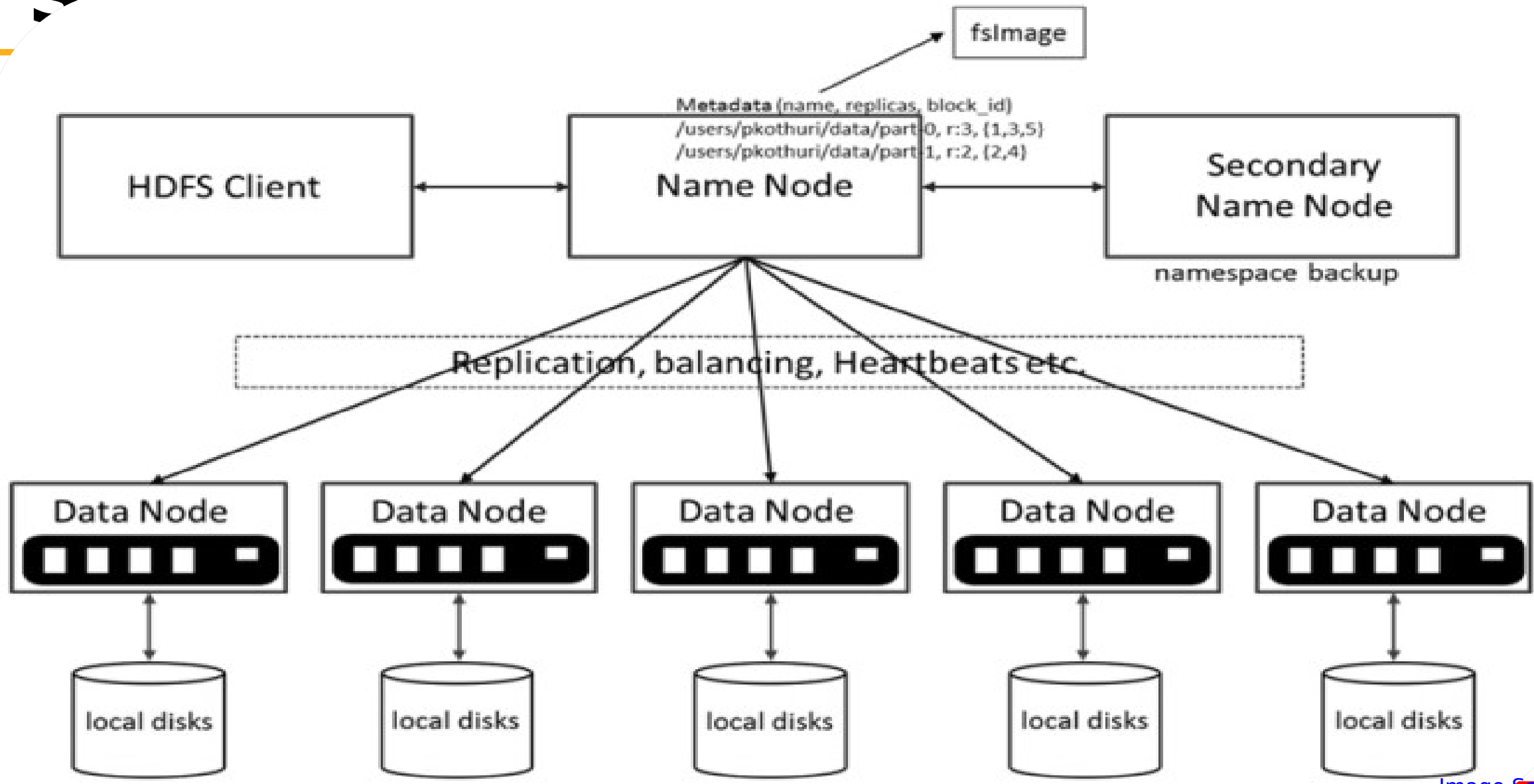


Image Sourc

Act, 1956

Hadoop Distributed File System - Contd

- based on Google File System (GFS) and was initially engineered to process data with the MapReduce programming model
- similar to object storage but with a key difference: Hadoop combines compute and storage on the same nodes, where object stores typically have limited support for internal processing

Management

- breaks large files into blocks, chunks of data less than a few hundred megabytes in size
- managed by the NameNode, which maintains directories, file metadata, and a detailed catalog describing the location of file blocks in the cluster
- each block of data is replicated to three nodes - increases both the durability and availability of data

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Hadoop Distributed File System - Contd

Claims that Hadoop is dead! This is only partially true!

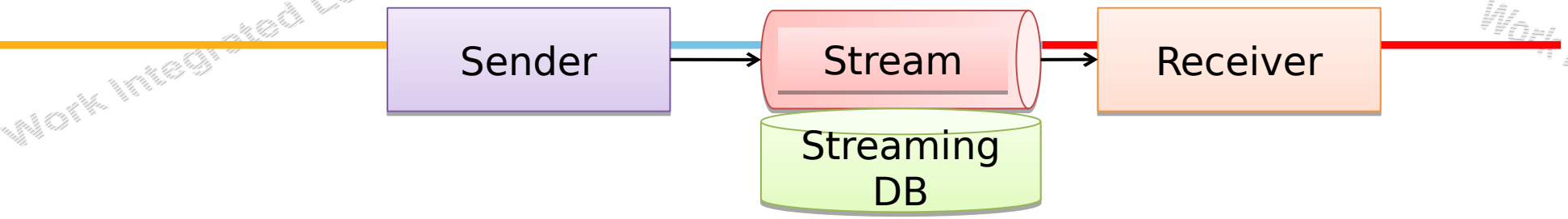
- **no longer a hot, bleeding-edge technology**
- **Many Hadoop ecosystem tools such as Apache Pig are now on life support and primarily used to run legacy jobs**
- **The pure MapReduce programming model has fallen by the wayside**

But HDFS remains widely used in various applications and organizations

- **HDFS is a key ingredient of many current big data engines, such as Amazon EMR**
- **In fact, Apache Spark is still commonly run on HDFS clusters**

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Streaming Data Storage



Streaming data has different storage requirements than non-streaming data - retention, replay!

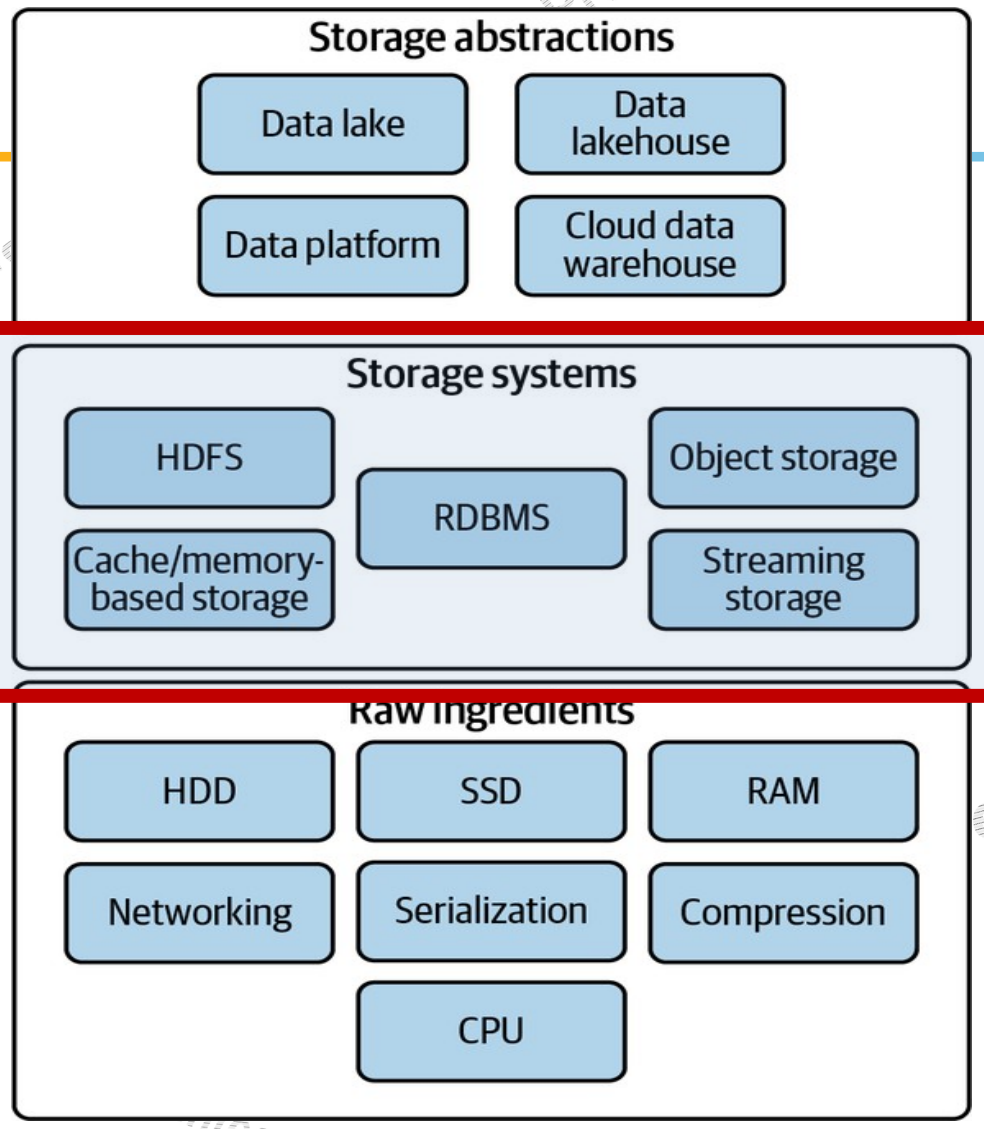
Retention

- **In the case of message queues, stored data is temporal and expected to disappear after a certain duration**
- **Distributed, scalable streaming frameworks like Apache Kafka now allow extremely long-duration streaming data retention**
- **Kafka competitors (including Amazon Kinesis, Apache Pulsar, and Google Cloud Pub/Sub) also support long data retention**

Replay

- **allows a streaming system to return a range of historical stored data**
- **standard data-retrieval mechanism for streaming storage systems**
- **can be used to run batch queries over a time range or to reprocess data in a streaming pipeline**

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956



Data Storage Infrastructure -Storage Abstractions

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Abstractions

- Data engineering storage abstractions are data organization and query patterns that sit at the heart of the data engineering lifecycle
- are built atop the data storage systems

The main types of abstractions are those that support data science, analytics, and reporting use cases

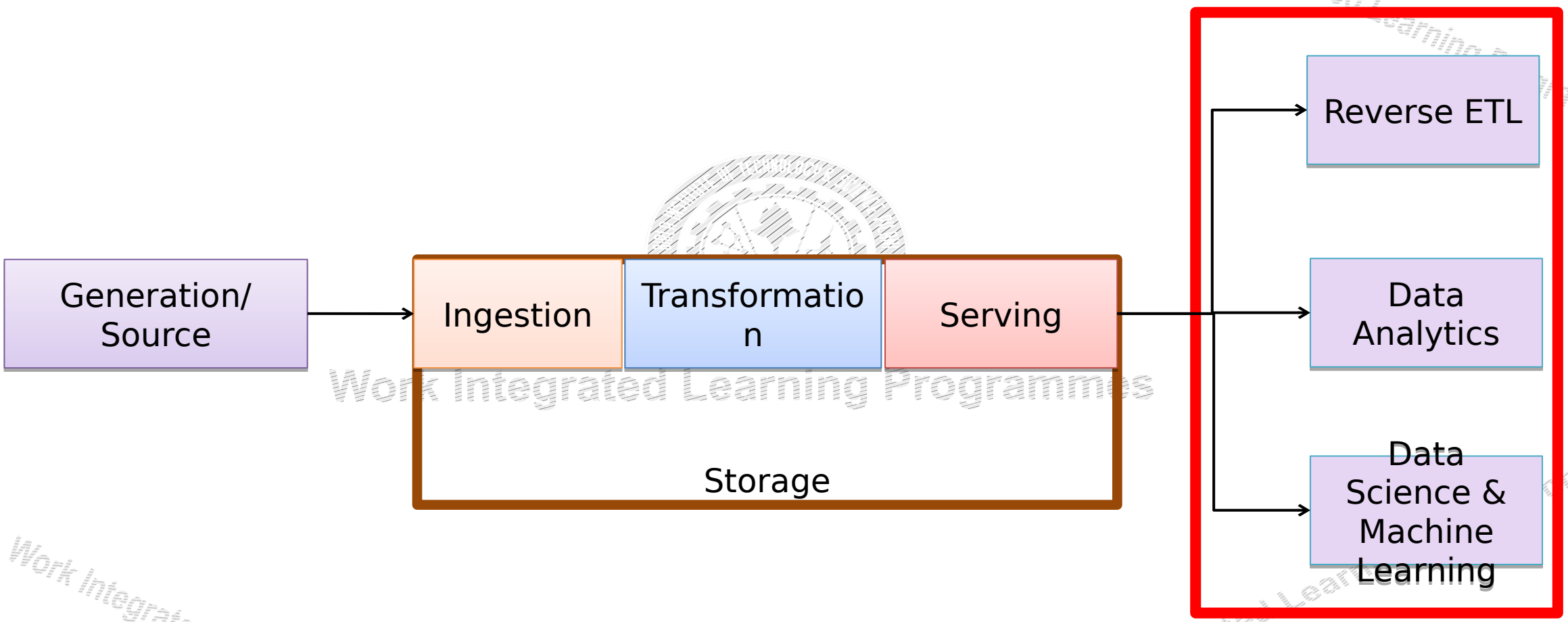
- Data warehouse
- Data lake
- Data lakehouse
- Data platforms
- Data catalogs

Work Integrated Learning Programmes

Data Serving Infrastructure

Work Integrated Learning Programmes

Introduction



BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Data Serving Infrastructure - Reverse ETL

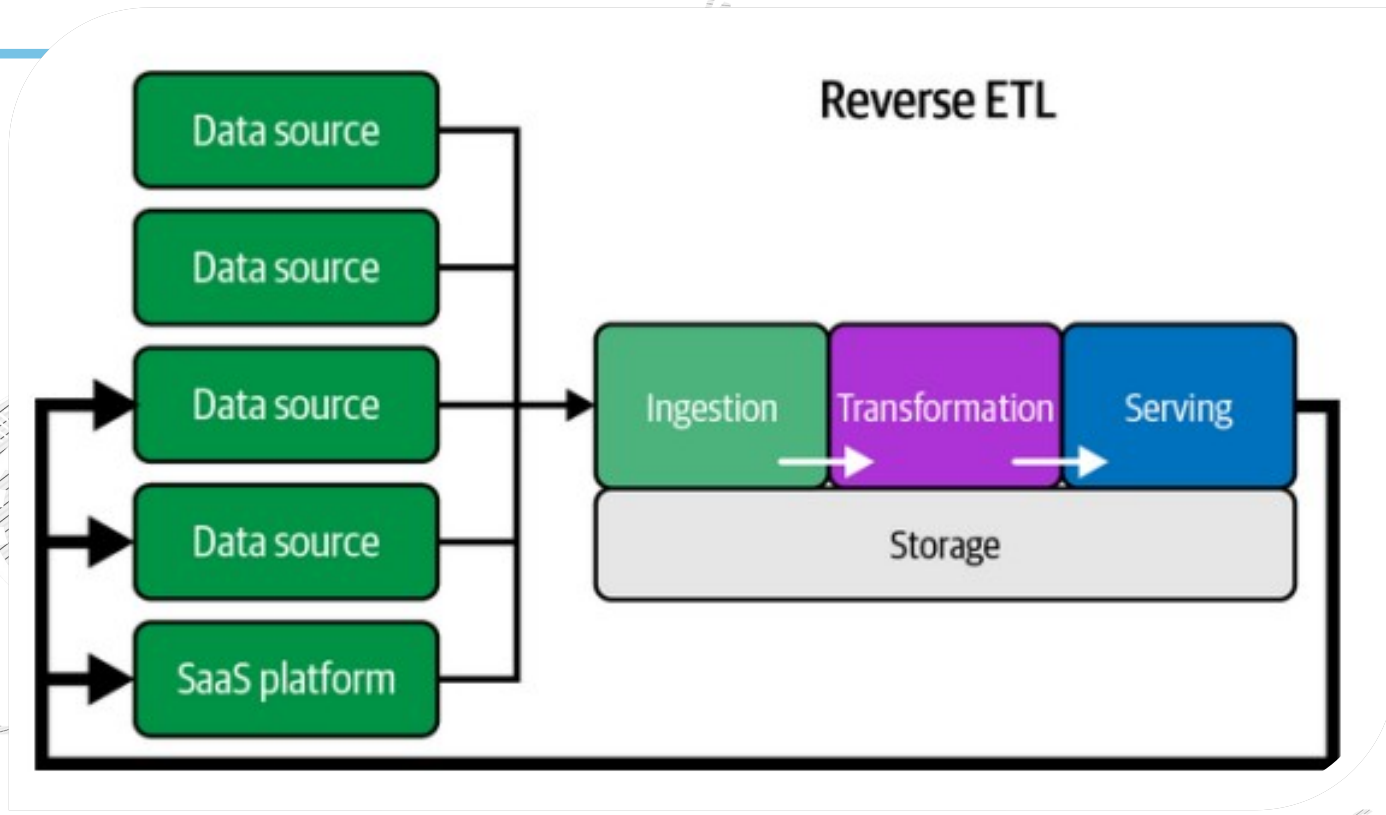
Work Integrated Learning Programmes

What is it?

- How many times, you feel that the latest business parameters and threshold if available as input will be helpful?

- process of sending data back to data sources
- For example, might acquire data from an adtech platform, run a statistical process on this data to determine cost-per-click bids, and then feed this data back into the ad tech platform

- Can you give an example?



BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Example

For example,

- A data engineer might pull customers and order data from a CRM and store it in a data warehouse
- This data is used to train a lead scoring model, whose results are returned to the data warehouse
- Company's sales team wants access to these scored leads to try to generate more sales

- Have a few options to get the results of this lead scoring model into the hands of the sales team
- Can put the results in a dashboard for them to view
- Or might email the results to them as an Excel file

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Detailed

- The challenge with these approaches is that they are not connected to the CRM, where a salesperson does their work
- Why not just put the scored leads back into the CRM?

- Using reverse ETL and loading the scored leads back into the CRM is the easiest and best approach for this data product
- Reverse ETL takes processed data from the output side of the data engineering lifecycle and feeds it back into source systems

- Do you see any challenges?
- A practical challenge in implementing this: Data transformation and validation from the output to the source system

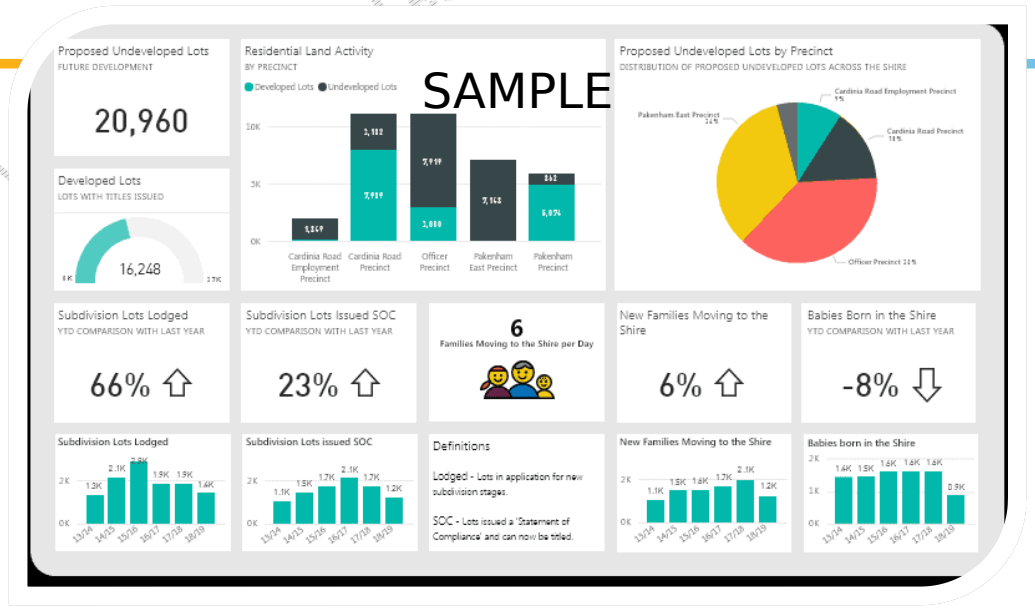
- Some tools used: Hightouch, Census, Fivetran, Airbyte, and RudderStack

Work Integrated Learning Programmes

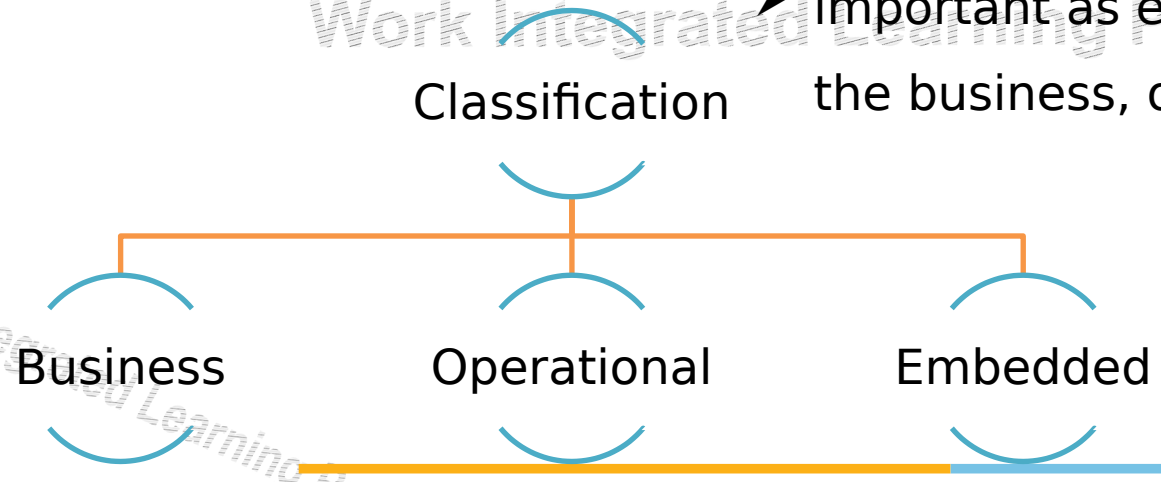
Data Serving Infrastructure - Data Analytics

Work Integrated Learning Programmes

Introduction



- Discovering, exploring, identifying, and making visible key insights and patterns within data
- Carried out using statistical methods, reporting, BI tools, and more
- most traditional area of data serving
- prepare data for use in statistical analysis, reporting, and dashboards
- important as ever for stakeholders to have visibility into the business, organizational, and financial processes



BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Detailed

innovate achieve lead

Business Analytics

- uses historical and current data to make strategic and actionable decisions
- types of decisions tend to factor in longer-term trends and often involve a mix of statistical and trend analysis, alongside domain expertise and human judgment
- typically falls into a few big areas—dashboards, reports, and ad hoc analysis

Operational Analytics

- uses data to take immediate action:
- Operational analytics versus business analytics = immediate action versus actionable insights
- big difference between operational and business analytics is time
- as real-time updates can be impactful in addressing a problem when it occurs

The line between business and operational analytics has begun to blur!

Sample tools: Tableu, PowerBI, Qlik

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Detailed

innovate achieve lead

Embedded Analytics

- Whereas business and operational analytics are internally focused, a recent trend is external-facing or embedded analytics
- With so much data powering applications, companies increasingly provide analytics to end users
- Typically referred to as data applications, often with analytics dashboards embedded within the application itself
- The landscape of embedded analytics is snowballing, and expect that such data applications will become increasingly pervasive within the next few years.

Sample tools: Looker, Qlik Embedded, Sisense

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Data Serving Infrastructure - Data Science & Machine Learning

Work Integrated Learning Programmes

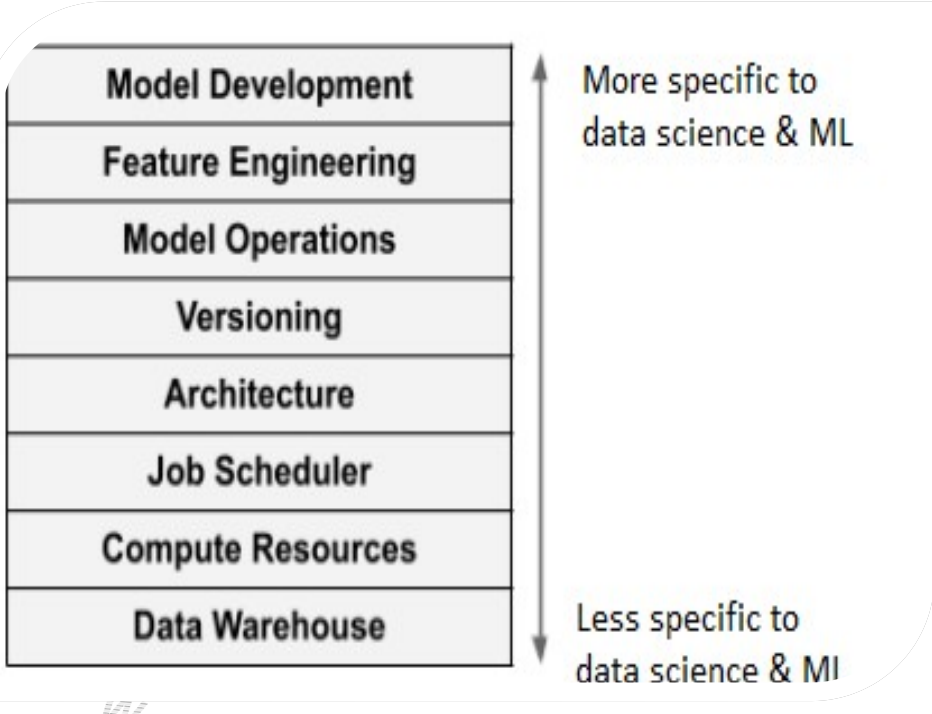
Introduction

All data science and machine learning projects have a few key elements in common:

- **In the technical point of view, all projects involve data and computation at their foundation.**
- **All projects will eventually need to address the question of integrating results into production systems which typically involves a great deal of software engineering.**
- **Finally, in the human point of view, all projects involve experimentation and iteration, which many consider to be the central activity of data science.**
- **Certainly possible for individuals companies or teams to come up with their own bespoke processes and practices to conduct data science projects,**
- **A common infrastructure can help to increase the number of projects that can be**
  - **executed simultaneously (volume),**
  - **speed up the time-to-market (velocity),**
  - **ensure that the results are robust (validity), and**
  - **make it possible to support a larger variety of projects.**

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

The Stack



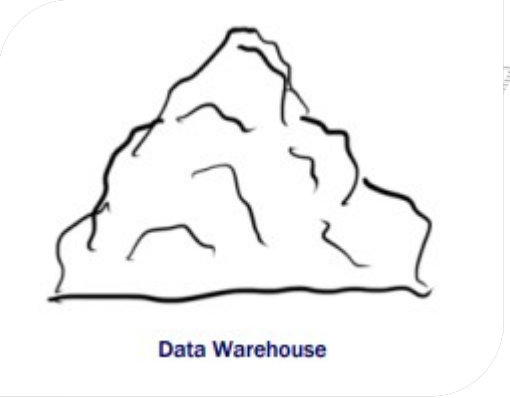
Because of culture of open source and relatively free technical information sharing between companies,

- **been able to observe and collect common patterns in data science & ML projects and infrastructure components**
  - **implementation details vary, the major infrastructural layers are relatively uniform across a large number of projects**
  - **each layer can be implemented in various ways, driven by the specific needs of their environment and use cases**
- **but the big picture is remarkably consistent!**
- **The most fundamental, generic components are at the bottom of the stack**
- **The layers become more specific to data science & ML towards the top of the stack**

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Datawarehouse

- The bottom most layer and we discussed about this multiple times
- Place where data is stored
- Doesn’t do anything by itself but how data is stored is critical
- A key challenge is to lay out the data in a way that makes it easily
- discoverable and efficiently accessible by data science applications while
- maintaining a high degree of durability (naturally the data warehouse must not lose data.)



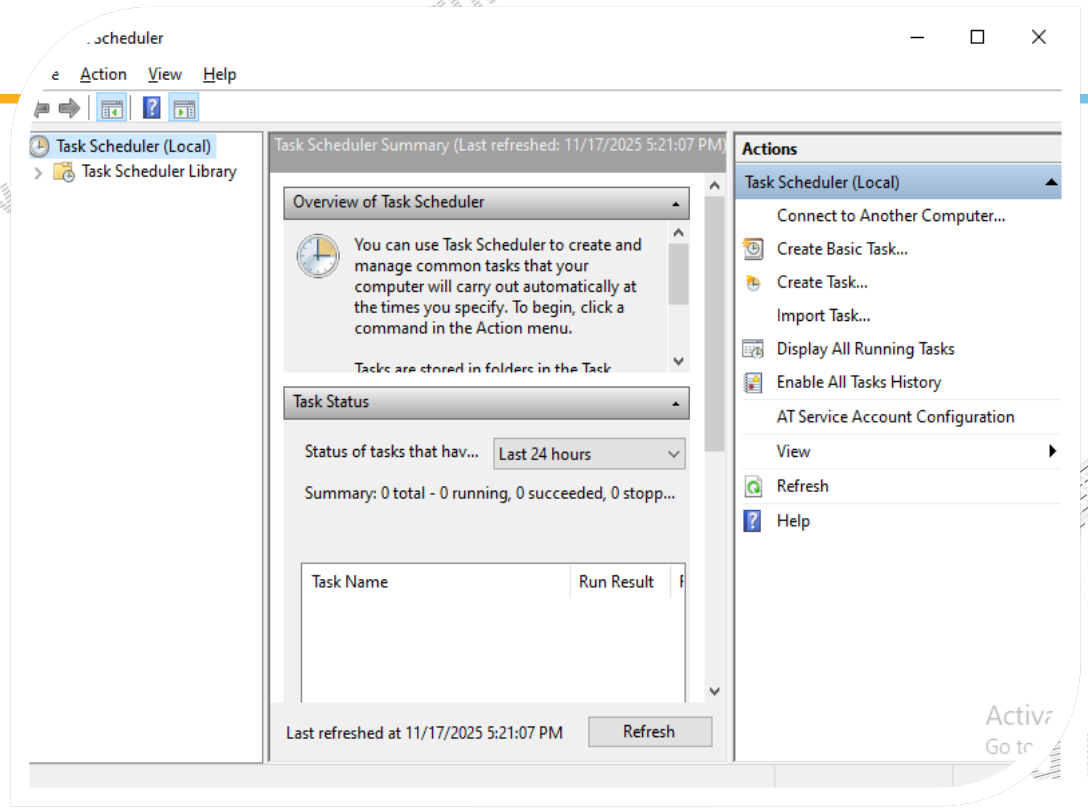
BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Compute Resources

- Need to have a way to do something with data, in other words, take some data, apply an algorithm and get results
- A challenge is that algorithms come in many shapes and sizes and need to run many of them, often in parallel, very frequently
- Need a system that can provide computing resources on demand
- The compute layer doesn't dictate what is computed
  - blindly accepts an algorithm from the outside world and executes it, consuming and producing data

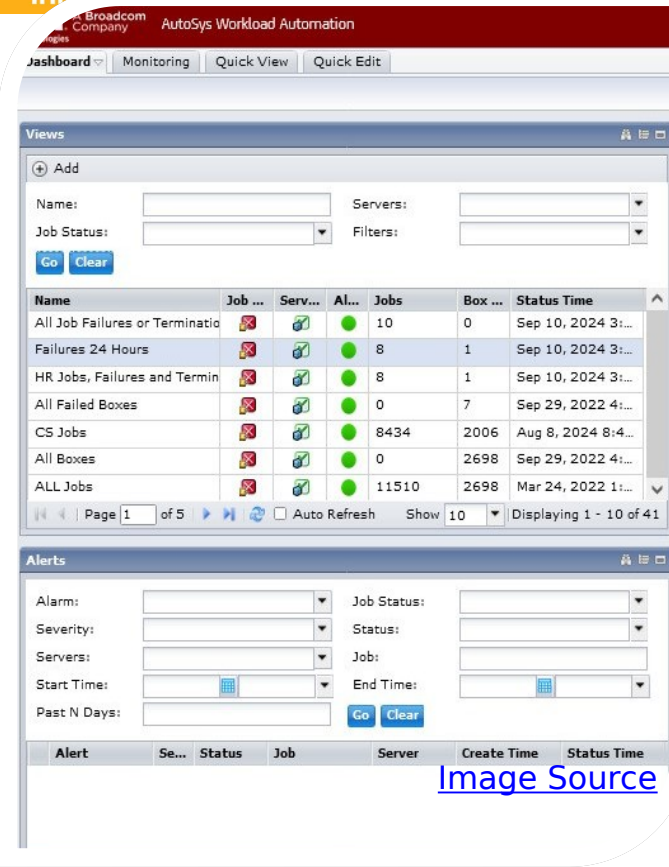
Work Integrated Learning Programmes

Job Scheduler


Task Scheduler
Action View Help
Task Scheduler (Local)
Task Scheduler Library
Task Scheduler Summary (Last refreshed: 11/17/2025 5:21:07 PM)
Overview of Task Scheduler
You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the times you specify. To begin, click a command in the Action menu.
Task Status
Status of tasks that hav...
Last 24 hours
Summary: 0 total - 0 running, 0 succeeded, 0 stopp...
Task Name
Run Result
Last refreshed at 11/17/2025 5:21:07 PM
Refresh
Actions
Task Scheduler (Local)
Connect to Another Computer...
Create Basic Task...
Create Task...
Import Task...
Display All Running Tasks
Enable All Tasks History
AT Service Account Configuration
View
Refresh
Help

Have you seen these screens?

Job Scheduler

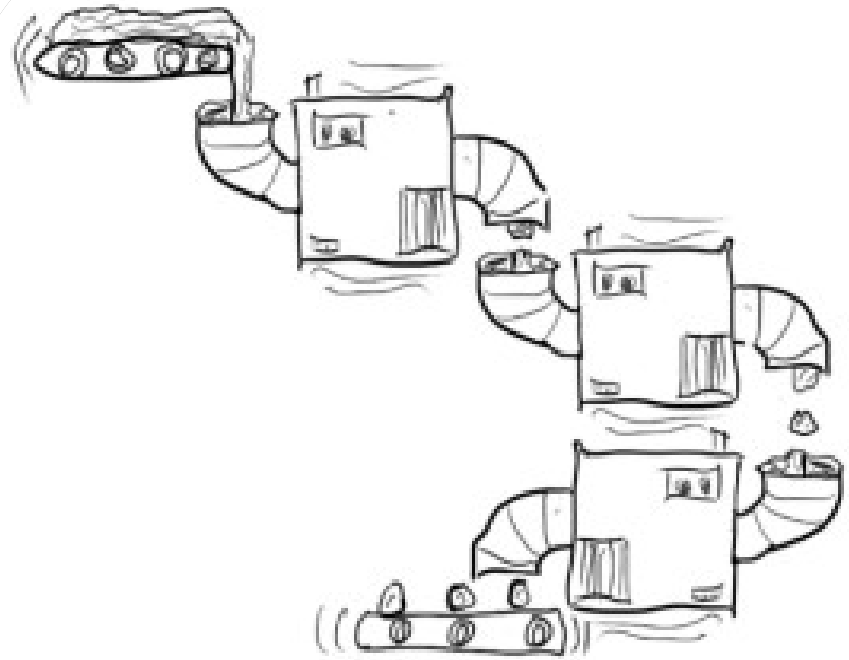


[Image Source]

- **Training a machine learning model or other data science application is rarely a one-time operation**
  - **More often it is a repeating process**
  - **Need to feed fresh data in the algorithm at a regular cadence**
- **Job of the scheduling layer to keep all the data flowing and all the machines humming uninterruptedly**
- **A key challenge is to keep the scheduling system itself up all the time and make sure it can manage almost any number of concurrent machines and data pipelines**

Architecture

- Data warehouse, compute resources, and job scheduler, are generic in nature: can be used to execute any workloads
- At the architecture layer defines what application is and how it is going to work
  - includes defining an algorithm or several algorithms and how the scheduling layer is supposed to connect the algorithms and the data pipelines together
- A key challenge is that of software engineering in general
  - How to map a real-world problem into an effective and robust software architecture.

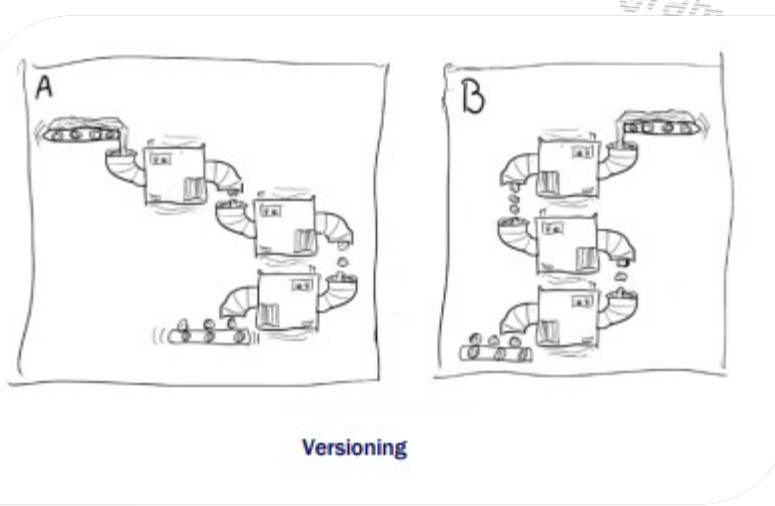


Architecture

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Versioning

- "experimentation and iteration" as one of the common themes in data science projects
- A concrete ramification of this is that architectures we define are always subject to change, makes sense to organize the work accordingly
- To complicate matters further, in the case of data science and/or ML the question of which version of the application is better is often non-trivial
- To judge the versions properly, you may need to run multiple versions side by side.
- A key challenge is that can evaluate results reliably only if the applications are well isolated, that is, they must not interfere with each other
- Must isolate the algorithms themselves, the input and output data, and the scheduled pipelines

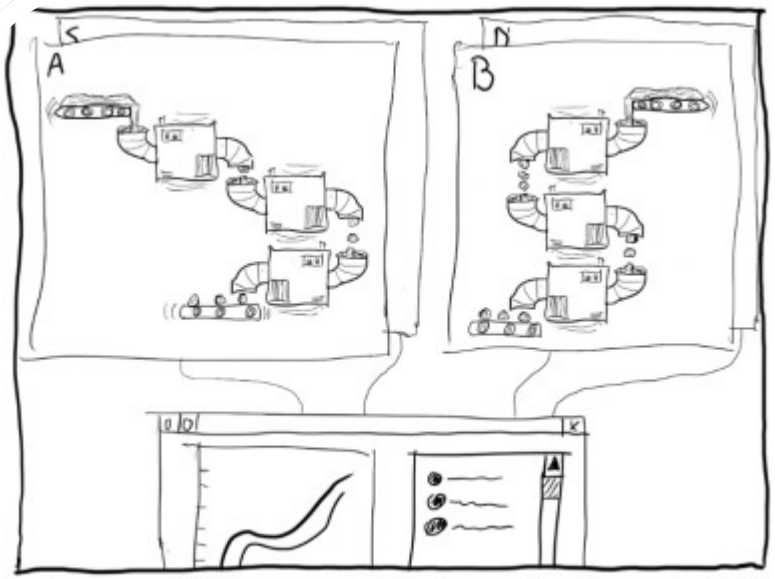


Versioning

Work Integrated Learning Programmes

Model Operations

- Imagine four versions of an application running, each with their own version of code and data
- Each version may be built by a separate data scientist, all of whom are all iterating on their versions independently.
- The result, unless organized properly, can be a logistical nightmare!
- Role of the model operations layer to help
  - to deploy, monitor, and assure validity of all models at all times, without hindering the speed of experimentation
  - To make this possible, the infrastructure needs to track metadata about all executions and models, from prototype to production

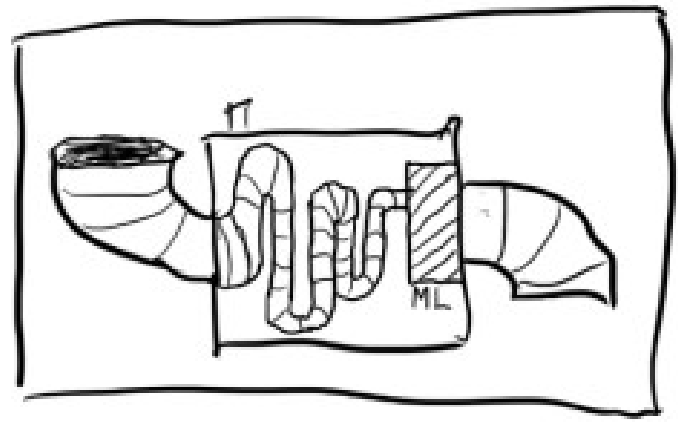


Model Operations

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Feature Engineering

- Typically, the raw input data is not fed directly into the model but it goes through a sequence of data transformations
- Designing this sequence is a major part of the data scientist's work when they are developing a new model or trying to improve performance of an existing model
- A key challenge is to make this process as efficient as possible



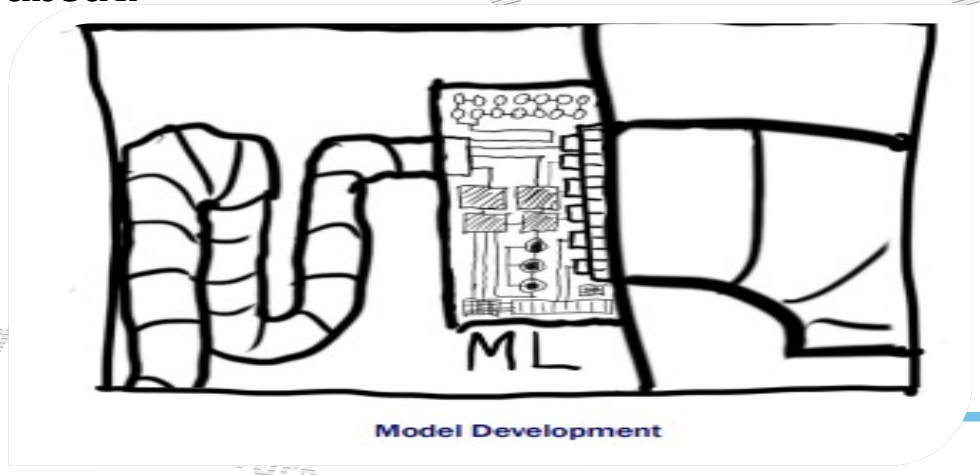
Feature Engineering

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Model Development

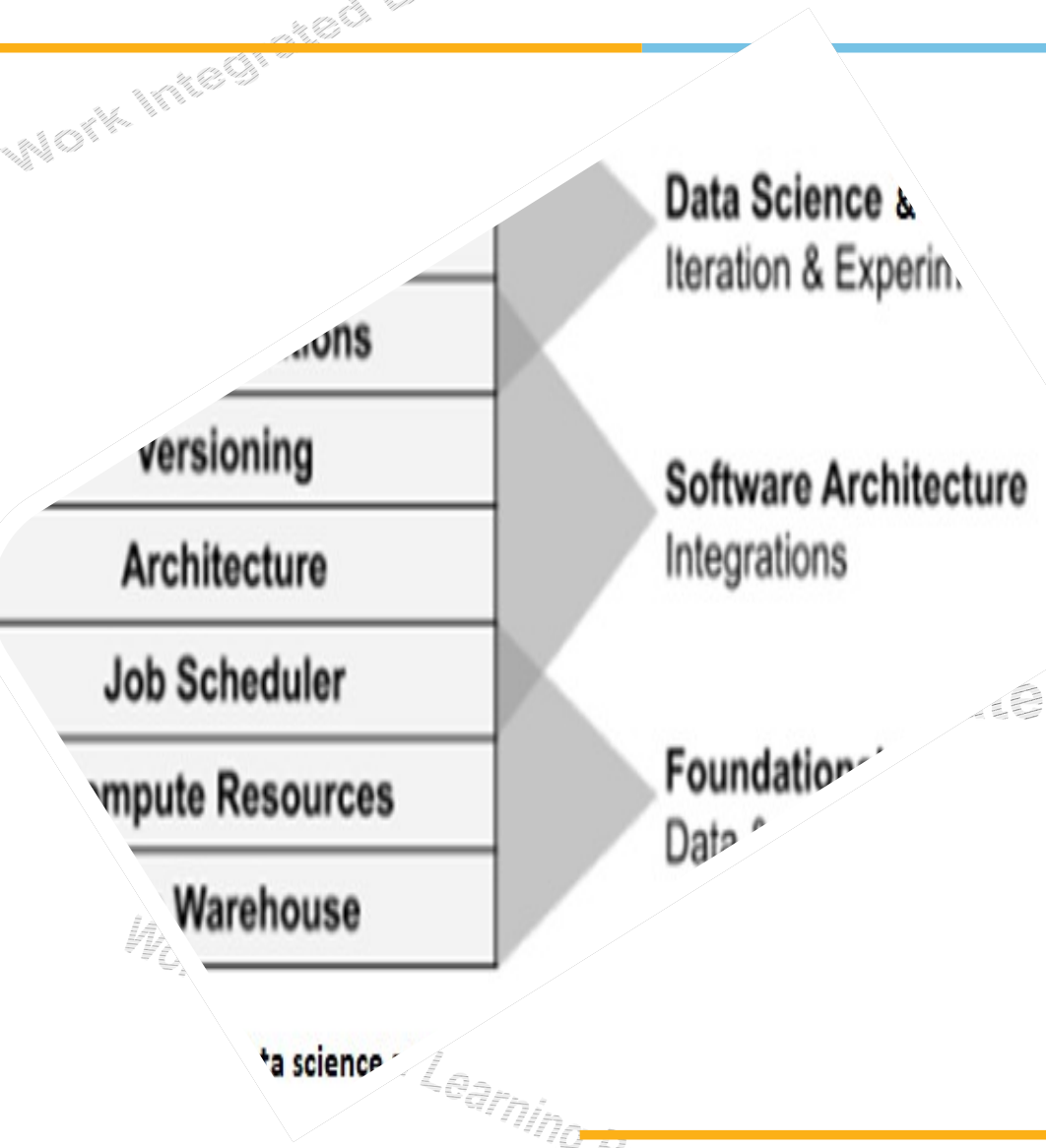
At the very top of the stack there's the layer of model development

- **finding and describing a mathematical model that transforms features into desired outputs**
- **A key challenge, and the key expertise of a data scientist, is to understand what modeling approach is most appropriate for the task at hand and how to implement it typically using off-the-shelf libraries.**
- **Occupies only a tiny part of the end-to-end machinery that makes an effective data science and/or ML application**
- **shouldn't underestimate the effort that it takes to embed intelligence in the surrounding environment, which is what this stack is all about!**



Work Integrated Learning Programmes

Stack Again



- Every data science project regardless of the problem domain needs to deal with data and compute
  - so these layers form the foundational infrastructure
  - layers are agnostic of what exactly gets executed

- The middle layers
  - define the software architecture of an individual data science or ML application
  - What gets executed and how: the algorithms, data pipelines, deployment strategies, and how results should be distribute
  - Much about the work is about integrating existing software components together.

- The top of the stack
  - realm of data science and or ML:
  - Defining a mathematical model and how to transform raw input to something that the model can process
  - can evolve quickly as the data scientist

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Data Scientists/ ML Engineers Freedom and Responsibility

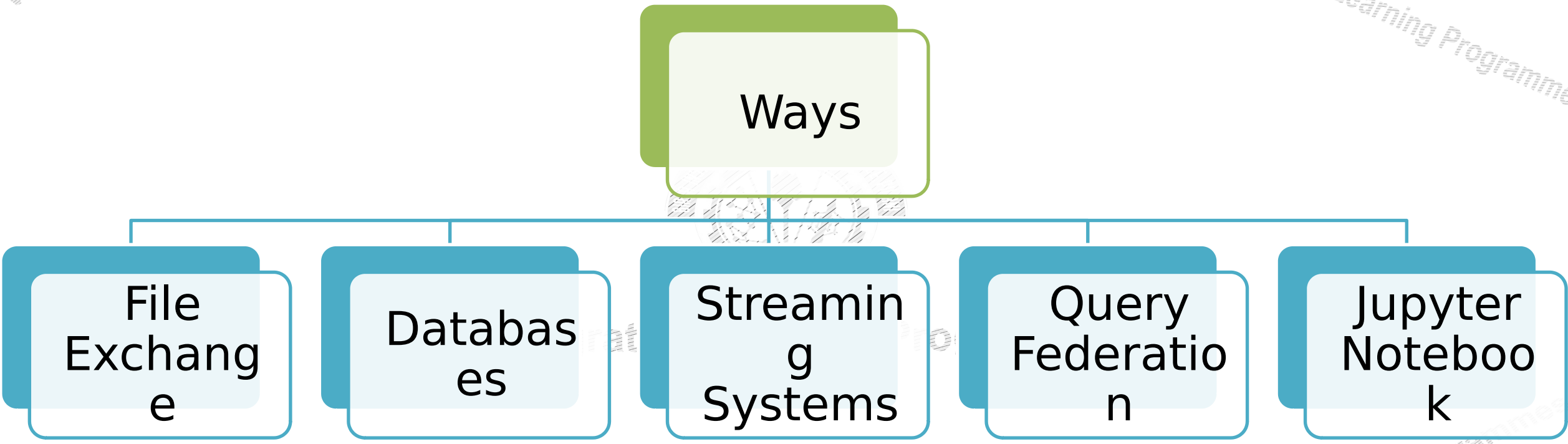
Infrastucture Complements the interests of a data scientist or ML Engineer

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Data Serving Infrastructure - Ways to serve data

Work Integrated Learning Programmes

Introduction



BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

File Exchange

- **Is ubiquitous in data serving. We process data and generate files to pass to data consumers**
  - A data scientist might load a text file (unstructured data) of customer messages to analyze the sentiments of customer complaints
  - A business unit might receive invoice data from a partner company as a collection of CSVs (structured data)
  - An analyst must perform some statistical analysis on these files

- **The way files are served depends on several factors, such as these:**
  - Use case — business analytics, operational analytics, embedded analytics
  - The data consumer's data-handling processes
  - The size and number of individual files in storage
  - Who is accessing this file
  - Data type — structured, semi-structured, or unstructured

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Databases

- Critical layer in serving data for analytics and ML - serving data from OLAP databases (e.g.,data warehouses and data lakes)
  - Serving data involves querying a database and then consuming those results for a use case
  - An analyst or data scientist might query a database by using a SQL editor and export those results to a CSV file for consumption by a downstream application, or analyze the results in a notebook
- Carries a variety of benefits
  - A database imposes order and structure on the **data through schema**
  - Databases can offer **fine-grained permission controls** at the table, column, and row level, allowing database administrators to craft complex access policies for various roles
  - Databases can offer **high serving performance** for large, computationally intensive queries and high query concurrency
- A data scientist might connect to a database, extract data, and perform feature engineering and selection
  - Converted dataset is then fed into an ML model
  - offline model is trained and produces predictive results
  - Data engineers are quite often tasked with managing the database-serving layer.

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Streaming Systems

- **Increasingly important in the realm of serving**
  - type of serving may involve emitted metrics, which are different from traditional queries
- **Operational analytics databases playing a growing role in this area**
  - allow queries to run across a large range of historical data, encompassing up-to-the-second current data
  - combine aspects of OLAP databases with stream-processing systems
  - increasingly work with streaming systems to serve data for analytics and ML, so get familiar with this paradigm

Work Integrated Learning Programmes

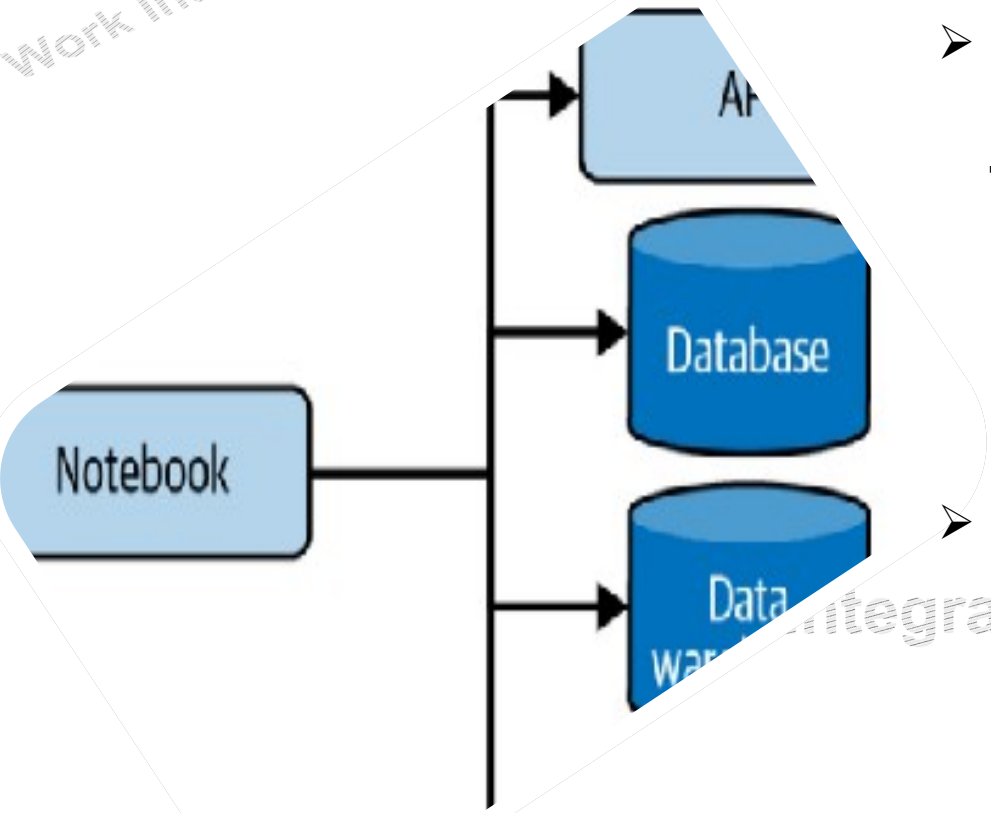
Query Federation

- Query federation pulls data from multiple sources such as data lakes, RDBMSs, and data warehouses
  - becoming more popular as distributed query virtualization engines gain recognition as ways to serve queries
  - without going through the trouble of centralizing data in an OLAP system
  - Trino and Presto and managed services such as Starburst
- End user might be querying several systems—OLTP, OLAP, APIs, filesystems, etc.
  - Instead of serving data from a single system, now serving data from multiple systems, each with its usage patterns, quirks, and nuances
  - If federated queries touch live production source systems, need to ensure that the federated query won't consume excessive resources in the source
- Federation allows ad hoc queries for performing exploratory analysis, blending data from various systems without the complexity of setting up data pipelines or ETL
  - Federated queries also provide read-only access to source systems, which is great when you don't want to serve files,

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Jupyter Notebook

- Data scientists often use notebooks in their day-to-day work for exploring data, engineering features, or training a model



- Jupyter Notebook, along with its next-generation iteration, JupyterLab
  - open source and can be hosted locally on a laptop, on a server, or through various cloud-managed services
  - Jupyter stands for Julia, Python, and R
- Data scientists will programmatically connect to a data source, such as an API, a database, a data warehouse, or a data lake
- In a notebook, all connections are created using
  - the appropriate built-in or imported libraries to load a file from a filepath
  - connect to an API endpoint

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Automation in Data Infrastructure Management

Work Integrated Learning Programmes

Summary of Data Infrastructure Components

- Storage systems (data lakes, warehouses, lakehouses)
- Databases (RDBMS, NoSQL, time-series)
- Compute (ETL/ELT engines, clusters, serverless)
- Orchestration engines
- Integration components (APIs, messaging)

Work Integrated Learning Programmes

Key Areas for Automation

innovate achieve lead

Infrastructure Provisioning

- **Provisioning of infrastructure through automation scripts**
- **Have you ever requested for virtual machines? How quickly you normally get them?**
- **If you say quite fast, how is it achieved?**

Tools

- **IaC(Infrastructure as Code): Terraform, Pulumi, AWS CDK, Azure Bicep**
- **GitOps: ArgoCD, Flux**
- **Cloud-native deployment templates – Cloud formation(You will see how your lab environment gets setup quickly in your webinar)**

Key Areas for Automation

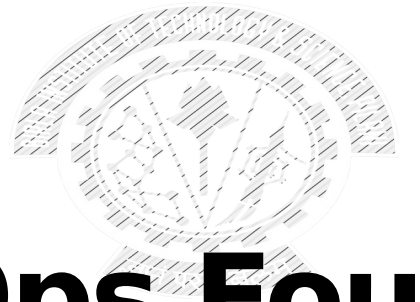
innovate achieve lead

Pipeline Automation

- **Key for data pipelines**
- **Includes ETL/ELT workflows, streaming jobs, trigger based transfers, scheduled data pipeline execution**
- **Automatically rerun failed tasks with backoff strategy**
- **Schema-change detection triggers auto-update of downstream models (e.g., dbt tests + deployments)**

Tools

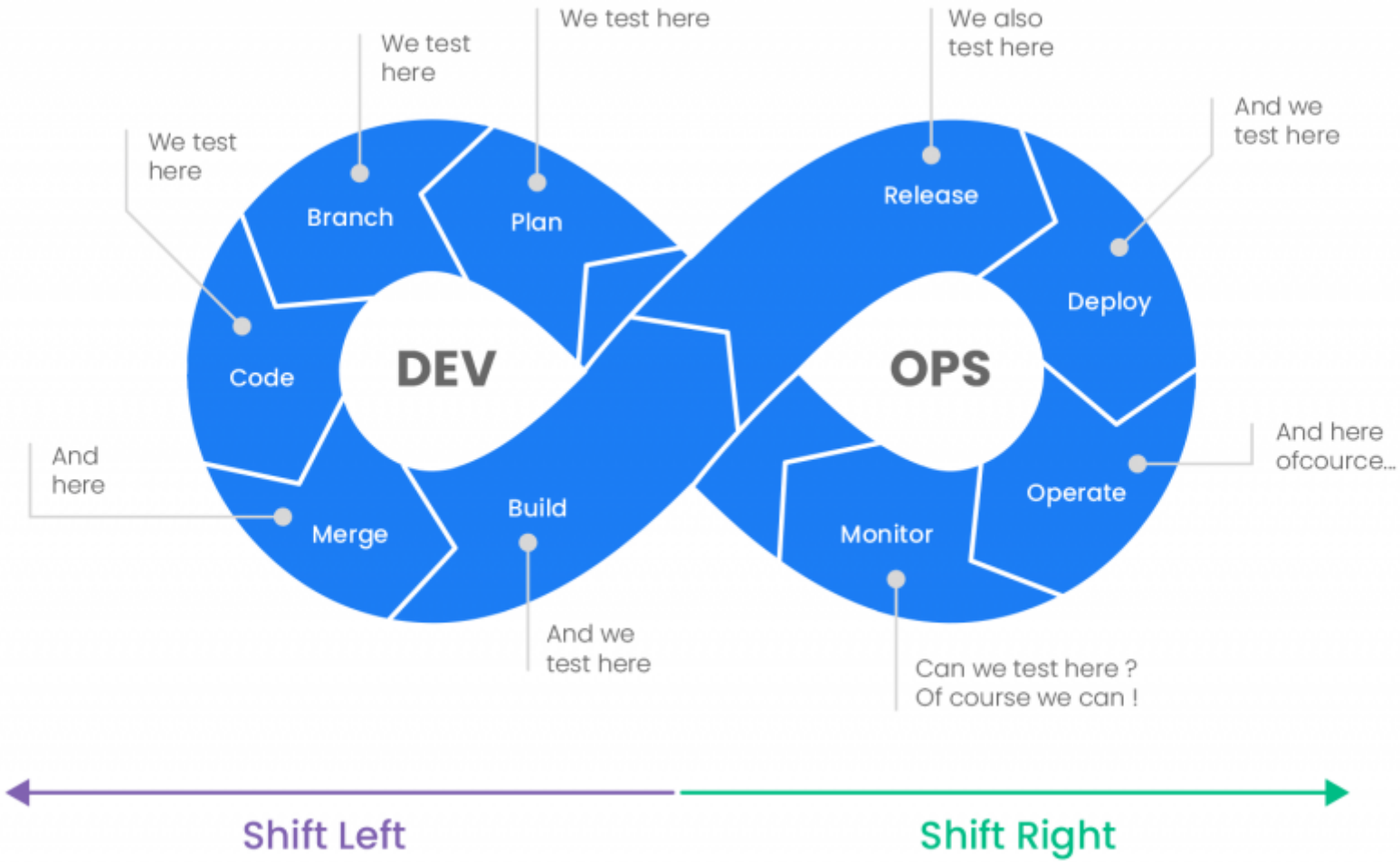
- **Orchestrators: Airflow, Dagster, Prefect**
- **Event-driven pipelines: EventBridge, Azure Event Grid, Kafka Streams**
- **Self-healing workflows (retry, fallback, alternative route)**



DevOps & DataOps Foundation

Work Integrated Learning Programmes

Devops



Here we work on code

Devops for data

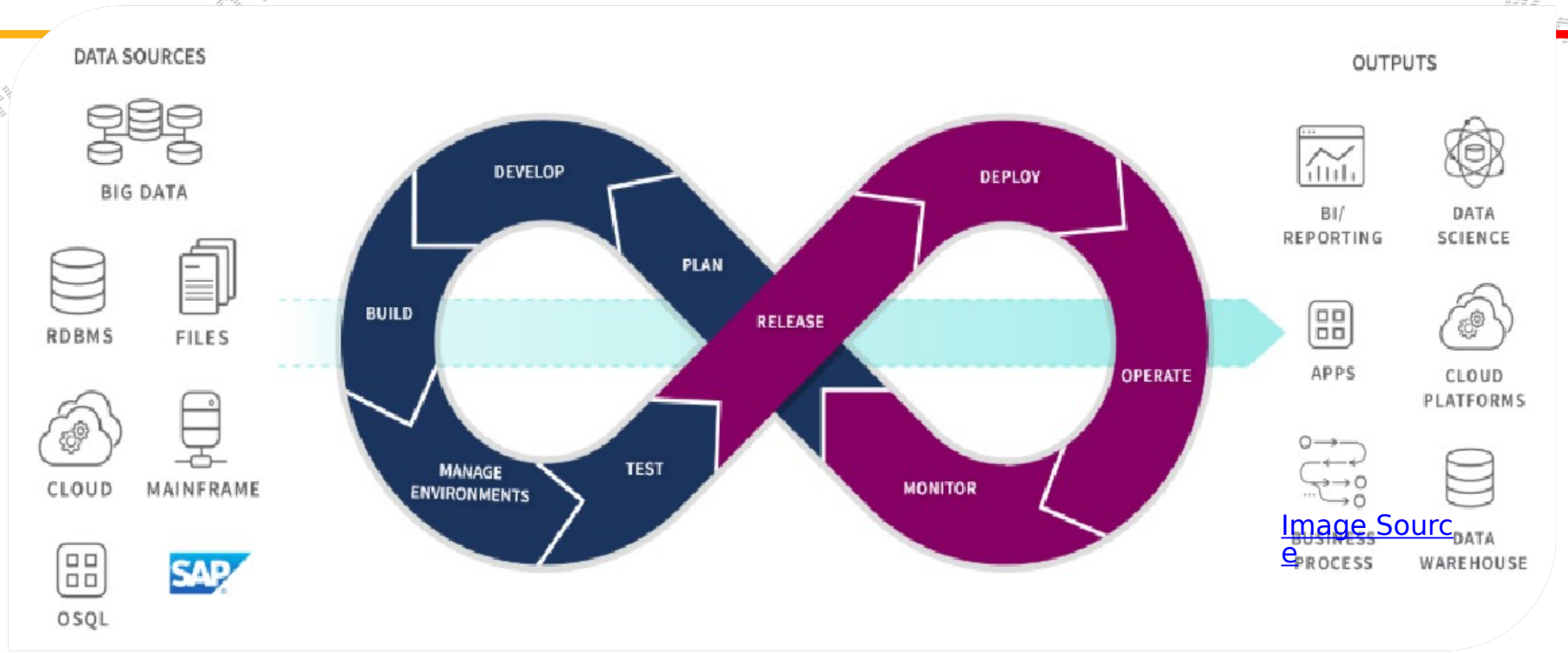


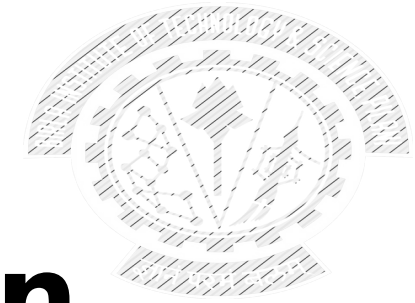
Image Source

Here we work on

- data pipeline development – which is dev
- Monitoring, troubleshooting, enhancing – which is ops

- DataOps employs agile processes for data governance and analytics development and DevOps processes for code optimization, product builds, and delivery. In addition to building new code, streamlining and improving the data warehouse are crucial.

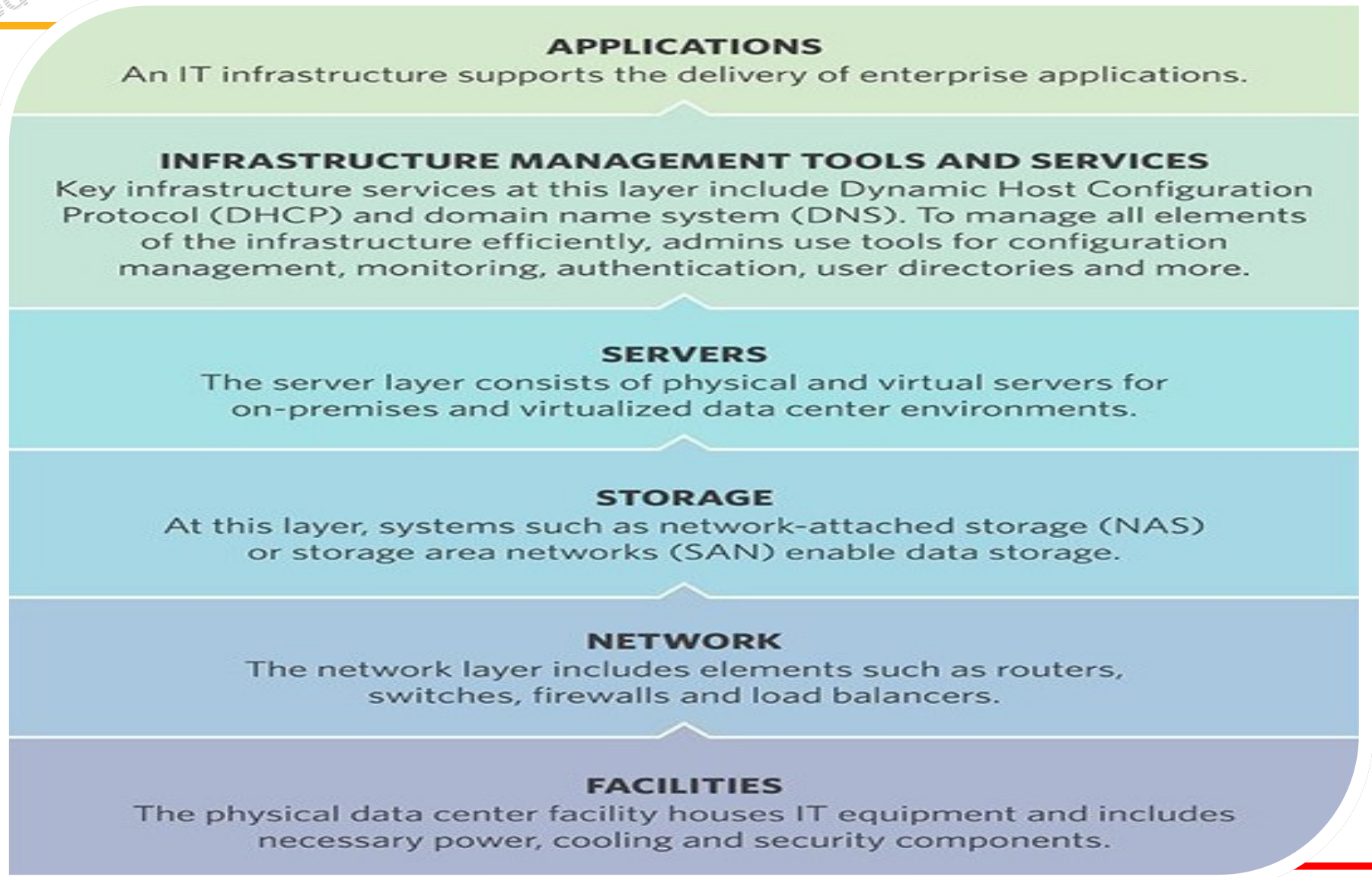
BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956



Cloud Foundation

Work Integrated Learning Programmes

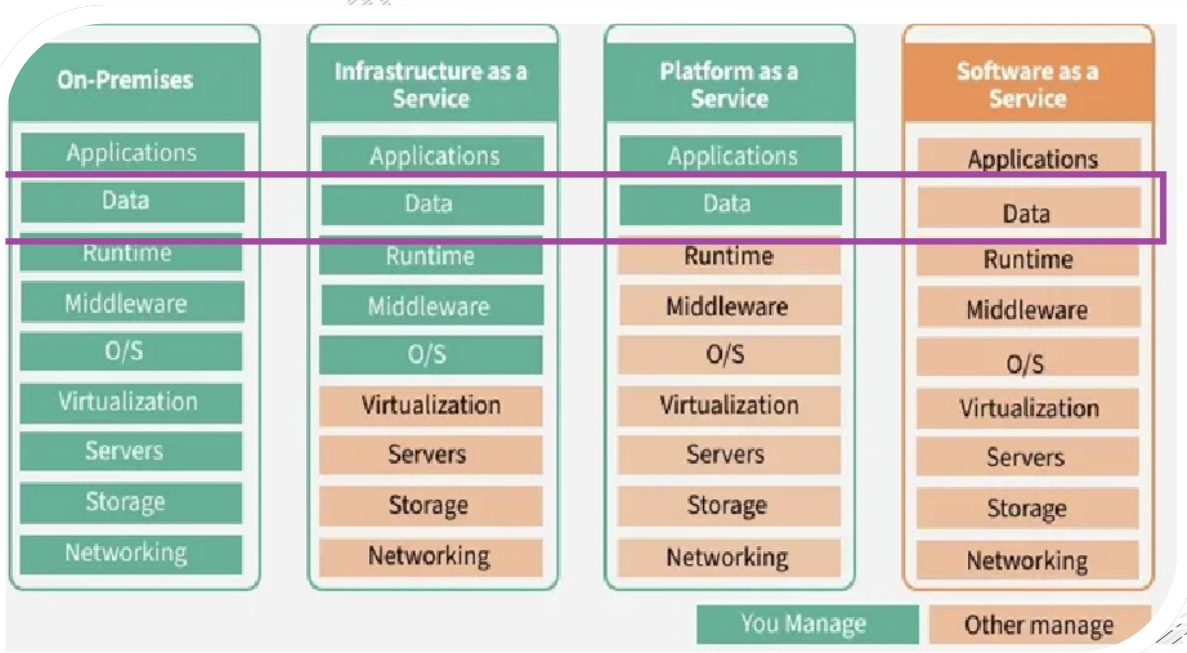
Typical On-Prem Infrastructure Components


APPLICATIONS
An IT infrastructure supports the delivery of enterprise applications.
INFRASTRUCTURE MANAGEMENT TOOLS AND SERVICES
Key infrastructure services at this layer include Dynamic Host Configuration Protocol (DHCP) and domain name system (DNS). To manage all elements of the infrastructure efficiently, admins use tools for configuration management, monitoring, authentication, user directories and more.
SERVERS
The server layer consists of physical and virtual servers for on-premises and virtualized data center environments.
STORAGE
At this layer, systems such as network-attached storage (NAS) or storage area networks (SAN) enable data storage.
NETWORK
The network layer includes elements such as routers, switches, firewalls and load balancers.
FACILITIES
The physical data center facility houses IT equipment and includes necessary power, cooling and security components.

Source

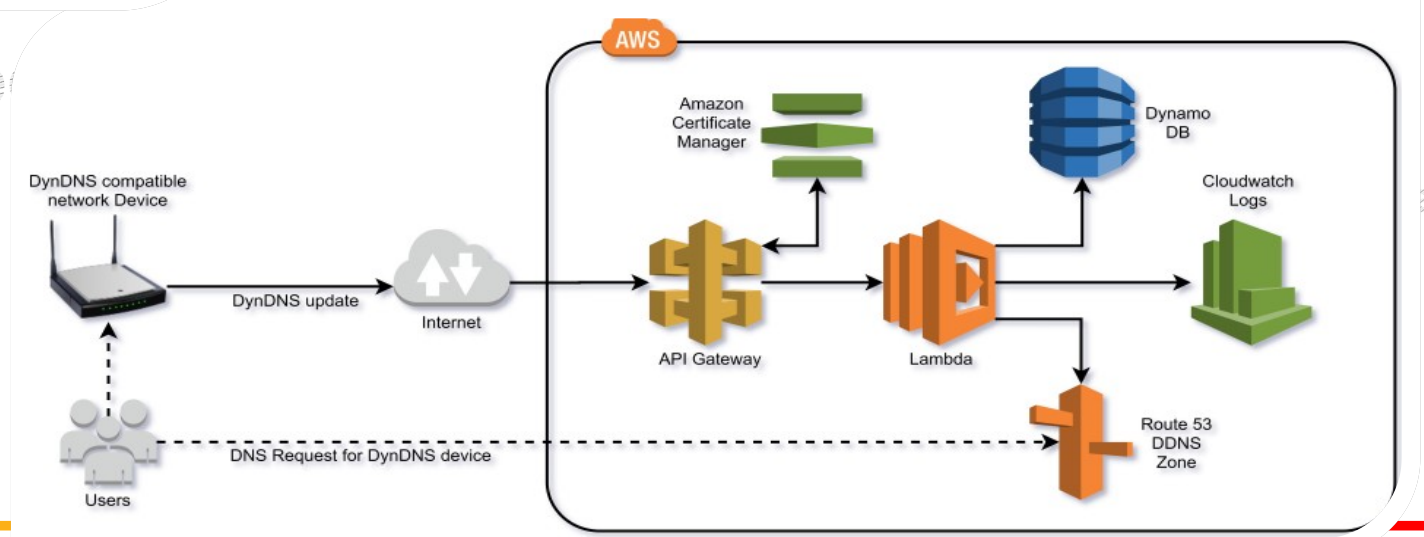
BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Cloud Infra



the delivery of computing services—including storage, databases, software, and servers—over the internet on a pay-as-you-go basis

Databases and its related services like, pipeline builders, streaming platforms, API gateways, data stores are readily available as a service



Security Foundation

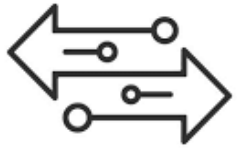
Work Integrated Learning Programmes

Data Security - Basics

What should we worry about?

Data At Rest

- Data that is stored in a medium
- Encryption of entire medium



Data At Transit

- Data that is actively moving from one system/application to another system/application
- Message Encryption

Data In Use

- Data that is being used by someone
- Role Based Access Control

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Continuous Training (CT) and Continuous Deployment (CD) pipelines

Work Integrated Learning Programmes

Introduction

CI / CD / CT for Machine Learning

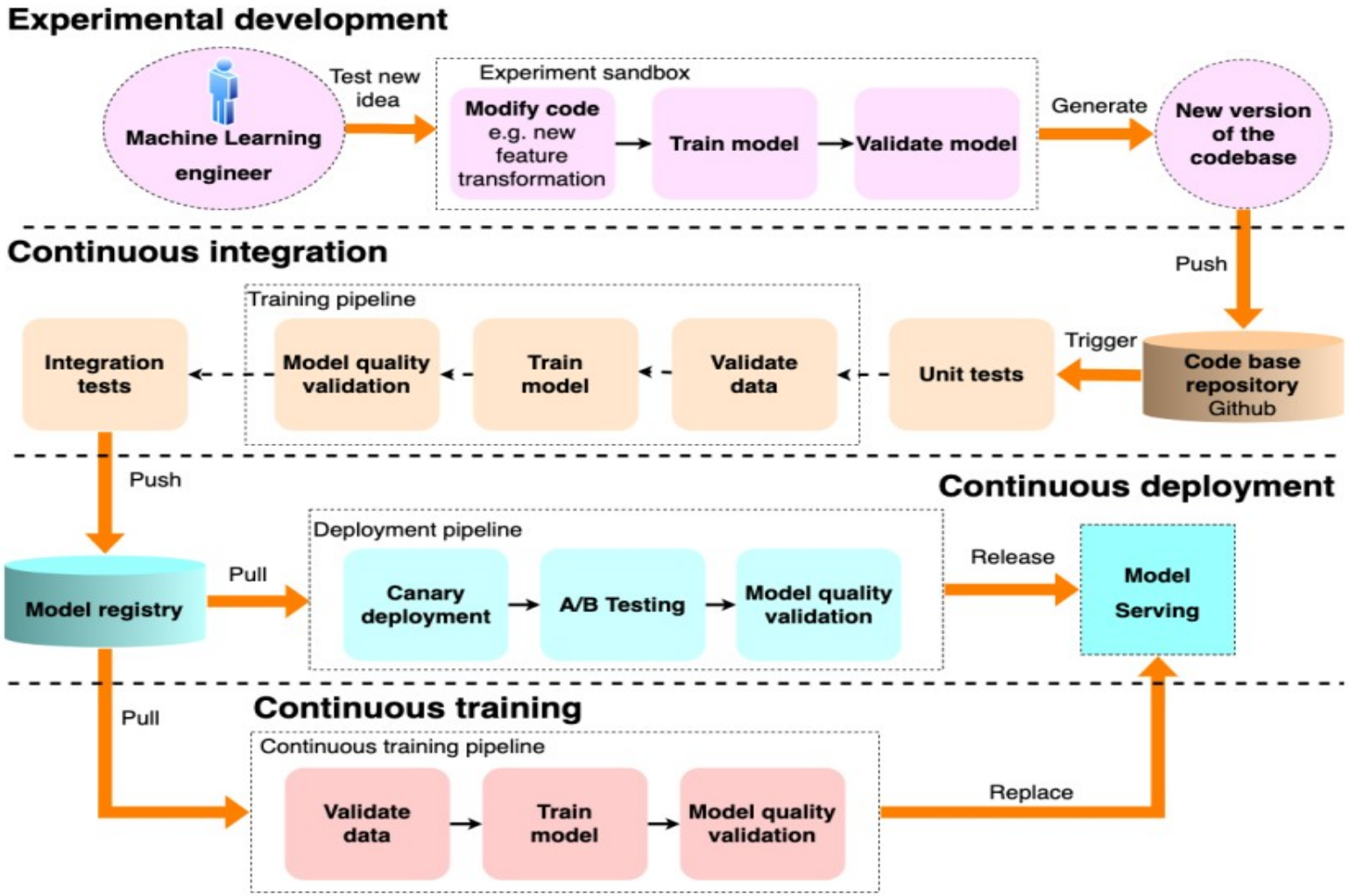


Image Source

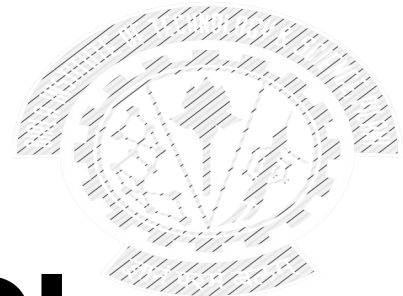
Key Phases

The experimental phase - The ML Engineer wants to test a new idea (let's say a new feature transformation). He modifies the code base to implement the new transformation, trains a model, and validates that the new transformation indeed yields higher performance. The resulting outcome at this point is just a piece of code that needs to be included in the master repo.

Continuous integration - The engineer then creates a Pull request (PR) that automatically triggers unit testing (like a typical CI process) but also triggers the instantiation of the automated training pipeline to retrain the model, potentially test it through integration tests or test cases and push it to a model registry. There is a manual process for another engineer to validate the PR and performance reading of the new model.

Continuous deployment - Activating a deployment triggers a canary deployment to make sure the model fits in a serving pipeline and runs an A/B test experiment to test it against the production model. After satisfactory results, we can propose the new model as a replacement for the production one.

Continuous training - as soon as the model enters the model registry, it deteriorates, and you might want to activate recurring training right away. For example, each day, the model can be further fine-tuned with the new training data of the day, deployed, and the serving pipeline rerouted to the updated model.



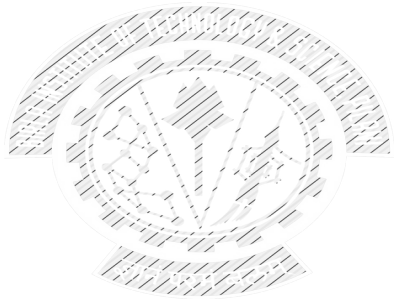
DataOps and Observability

Work Integrated Learning Programme

Key Phases

- Observability is no longer just for software engineering!
  - With the rise of data downtime and the increasing complexity of the data stack,
  - observability has emerged as a critical concern for data teams, too.
- Developer Operations (DevOps) teams have become an integral component of most engineering organizations
  - remove silos between software developers and IT, facilitating the seamless and reliable release of software to production
- As organizations grow and the underlying tech stacks powering them become more complicated
  - it's important for DevOps teams to maintain a constant pulse on the health of their systems
- Observability, a more recent addition to the engineering lexicon, speaks to this need, and refers to the monitoring, tracking, and triaging of incidents to prevent downtime.
- A dedicated session on observability will be coming in future

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956



Case Study

Work Integrated Learning Programmes

Background & Problem Statement

- DoorDash is an American technology company that operates an on-demand delivery platform connecting consumers with local merchants for food, groceries, and other retail items. It is the largest food delivery service in the United States and facilitates deliveries through independent contractors called "Dashers".

Business Problem

- Their earlier ML workflows were:
- Fragmented (analytics, batch ML, real-time ML all separate)
- Low observability (data issues → incorrect ETAs)
- Slow deployment (weeks)
- Frequent data drift
- Manual pipeline debugging
- DoorDash needed a unified, reliable, production-grade ML data platform.

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Architecture of the DoorDash ML Platform

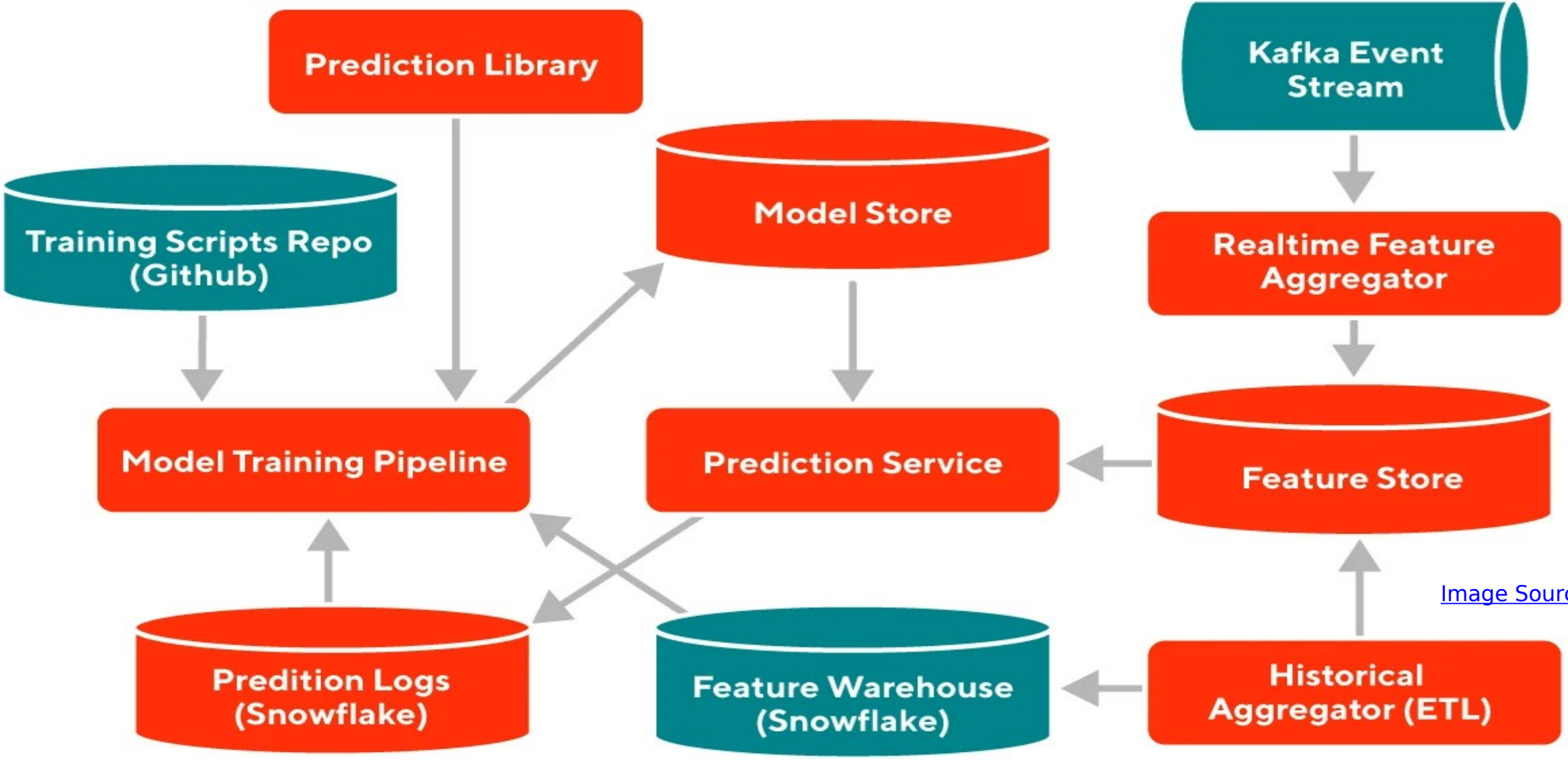


Image Source

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Components

Feature Store - Low latency store from which Prediction Service reads common features needed for evaluating the model. Supports numerical, categorical, and embedding features.

Realtime Feature Aggregator - Listens to a stream of events and aggregates them into features in realtime and stores them in the Feature Store. These are for features such as historic store wait time in the past 30 mins, recent driving speeds, etc.

Historical Aggregator - This runs offline to compute features which are longer-term aggregations like 1W, 3M, etc. These calculations run offline. Results are stored in the Feature Warehouse and also uploaded to the Feature Store.

Prediction Logs - This stores the predictions made from the prediction service including the features used when the prediction was made and the id of the model used to make the prediction. This is useful for debugging as well as for training data for the next model refresh.

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

Components - contd

Model Training Pipeline - All the production models will be built with this pipeline. The training script must be in the repository. Only this training pipeline will have access to write models into the Model Store to generate a trace of changes going into the Model Store for security and audit. The training pipeline will eventually support auto-retraining of models periodically and auto-deploy/monitoring. This is equivalent to the CI/CD system for ML Models.

Model Store - Stores the model files and metadata. Metadata identifies which model is currently active for certain predictions, defines which models are getting shadow traffic.

Prediction Service - Serves predictions in production for various use cases. Given a request with request features, context (store id, consumer id, etc) and prediction name (optionally including override model id to support A/B testing), generates the prediction.

BITS Pilani, Deemed to be University under Section 3 of UGC Act, 1956

References:

Data Pipelines Pocket Reference By James Densmore
Fundamentals of Data Engineering By Joe Reis and Matt Housley
Effective Data Science Infrastructure By Ville Tuulos
Summary from
https://medium.com/@ZhouAnnaWrites/book-notes-fundamentals-of-data-engineering-chapter-6-storage-81a533f81f0c

https://www.linkedin.com/pulse/continuous-integration-deployment-training-ci-cd-ct-damien-ajx7c/

https://www.techtarget.com/searchdatacenter/definition/infrastructure

https://www.pcloudy.com/cicd-pipeline-demystifying-the-complexities/

https://www.hardtechnology.net/post/dynamic-dns/
https://medium.com/innovationendeavors/operational-machine-learning-for-the-modern-data-stack-178a3feba334

Deemed to be University under Section 3 of UGC Act, 1956

Thank You!

Work Integrated Learning Programmes